MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 71 9683

ISSUE 6
MAY 1980

PRICE $2.50

		PAGE
ALL ABOUT MICRO-80		1
EDITORIAL		2
NEXT MONTH IN MICRO-80		4
BETTER BYTES		5
ASSEMBLY LANGUAGE PROGRAMMING (part 4)		6
GIVE YOUR '80 ON-BOARD SOUND		11
PROBLEM CORNER - ON THE MATTER OF GREMLINS		12
KEY REPEAT ON YOUR TRS-80		14
ON HITS AND GATS AND OTHER MYSTERIES		15
INPUT/OUTPUT (readers' letters)		15
SUB ATTACK	LI/4K	18
SPACE DRIVE	LI/16K	18
TRIG/BAS	LII/4K	18
SUPER SIZZLER	LII/16K	18
TIC-TAC-TOE	LII/4K	19
HOUSEHOLD ACCOUNTS	LII/16K	19
MICRO-80 PRODUCTS lift-out		23
CONNECTA and CONNECTX	LII/4K	25
LEVEL I SOFTWARE LISTINGS start		26
LEVEL II SOFTWARE LISTINGS start		32
CASSETTE EDITION INDEX		44
PRODUCT ORDER FORM		45
WHAT YOU MAY HAVE MISSED		45

MICRO-80 is registered for posting as a publication - CATEGORY B.

Printed by: The Shovel and Bull Printers,
312A Unley Road, Hyde Park, S.A. 5061.

All enquiries should be directed to:-
MICRO-80, P.O. BOX 213, GOODWOOD, SA 5034
Telephone (08) 71 9683

MICRO-80

** ABOUT MICRO-80 **

EDITOR: Ian Vagg
ASSOCIATE EDITORS: Peter Hartley, Eddy Paay.

MICRO-80 is the only Australian monthly magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80. It is available by subscription, $24.00 for 12 months or by mail order at $2.50 per copy. A cassette containing all the programs in each month's issue is available for an additional $3.50 or a combined annual subscription to both magazine and cassette, is available for $60.00. Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80 or System 80 and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

** WE WILL BUY YOUR PROGRAM **

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

** CONTENT **

Each month we publish at least one applications program in Level 1 BASIC, one in Level 2 BASIC and one in DISK BASIC (or disk compatible Level 2). We also publish Utility programs in Level 2 BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

** ADVERTISING **

We accept camera ready copy for display advertising at the following rates:

- FULL PAGE (19cm. wide x 24cm. high) $50.00
- 1/2 PAGE (19cm. wide x 12cm. high) $30.00

Classified advertisements are $5.00 for up to 50 words. Advertisements must be submitted by the 23rd of each month in order to appear in the following month's issue. A Company Order or payment must be included with the advertisement.

** COPYRIGHT **

** LIABILITY **

The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

-=* EDITORIAL *=-

If this is your fisrt copy of MICRO-80, "Welcome". We hope that you will enjoy reading our magazine, and that it will help you to get the most out of your TRS-80 or System 80. MICRO-80 is unique amongst Australian Magazines. To start with, it is written for a relatively small audience, those 6200 odd people who own TRS-80's and System 80's. Next, of course, most of the articles and programs published are written by our own readers, who receive a royalty payment in return. Finally, we believe that we are the most interactive technical magazine in existence. We react to readers' requests for articles; we publish readers' letters virtually in full, and we give meaningful answers. If you don't read the Letters to the Editor (Input/Output), then you are certainly missing out on a great deal of valuable advice and information.

Associated with MICRO-80 is MICRO-80 PRODUCTS, which publishes software on cassette and disk, and designs and manufactures hardware for '80 users. You will see MICRO-80 PRODUCTS' advertisements in each edition of MICRO-80, as well as in the technical press such as ETI. The software published by MICRO-80 PRODUCTS has all been written by readers of MICRO-80, who receive 35% of the gross sales of their programs as a royalty payment. 22.5% of sales is set aside for advertising and promotion, ensuring that most '80 users will quickly become aware of our readers' efforts. If you have written a program which you consider worthy of publication by MICRO-80 or by MICRO-80 PRODUCTS, send it in, together with the APPLICATION FORM which appears in every issue of MICRO-80. After all, you have nothing to lose except for the return postage, and you could earn yourself a lot of cash.

==*= SCOOP =*=*=*

** DICK SMITH ELECTRONICS - SYSTEM 80

The first 100 of the long-awaited System 80's were quietly landed at Sydney Airport on May 1st. By now there should be quite a few proud new owners learning all about their new machines. We believe that there are at least 500 firm orders for the System 80, and if it is as good as it sounds, then there should be a lot more than that sold in the very near future.

We extend our congratulations to the Dick Smith organisation for persevering with this project for so long. We understand that it took nearly 18 months to clear the licensing arrangements for the Microsoft BASIC interpreter, alone.

It is our belief that the increased competition within the Austalian market-place can only be of benefit to us all. Perhaps, now, Tandy's prices will come more into line with their U.S: prices (a Level I 4K machine costs $499 + sales tax (typically 6%) in America).

Needless to say, MICRO-80 will support the System 80 as well as the TRS-80.

** BETTER DISTRIBUTION FOR MICRO-80 AND MICRO-80 PRODUCTS

We have arranged for MICRO-80 to be sold from several selected retail outlets.
Dick Smith Electronics have taken a trial order of the April Edition for their Sydney outlets

Also in Sydney, CISA (Complete Information Systems of Australia) will stock each issue of MICRO-80, and will also sell MICRO-80 PRODUCTS software. CISA is at 159 Kent Street, opposite the IBM building, and specialises in products for the TRS-80.

In Brisbane, RYMAC SOFTWARE of 2 Balanda Street, Jindalee, will be stocking MICRO-80. Garry McKenzie, of RYMAC, is an ex Tandy Computer Centre Manager, who specialises in developing custom software for TRS-80's in business applications.

Over the next few months, we look forward to announcing a number of additional stockists of both MICRO-80 and MICRO-80 PRODUCTS software.

** OUR FREE SOFTWARE OFFER **

Emblazoned across the cover of this issue, is the first announcement of a free software cassette to be given to all new subscribers. The purpose of this offer is to boost our circulation to >2000 by the end of 1980. The cassette contains a selection of programs that we have published over our first six months. These are:-

Level I	
Hangman	Feb '80
Super Mastermind	Dec '79
Ricochet	Jan '80
Level II	
Household Accounts	May '80
Snake	Dec '79
Analogue Clock	Mar '80

If any of our existing SUBSCRIBERS would like a FREE copy of this cassette, just send us a small self-addressed padded post-bag with a 28c stamp on it, and we will send the cassette by return

** NEW PRODUCT NEWS

Last month we promised to give you information about an unnamed "accessory for your '80 that you wouldn't want to be without". Well, we've got TWO of them!

The first part of our Big News is that MICRO-80 PRODUCTS intends to be handle the EXATRON STRINGY FLOPPY. This is a high-speed sub-miniature cassette, especially designed to add-on to the TRS-80. The unit uses specially constructed cassettes (so small that they are called "wafers") holding an endless loop of digital tape. The wafers come in 10, 20, and 50 foot sizes, holding 8, 16 and 32K of data respectively. You can put multiple files on one wafer, and 16K on a 20 foot wafer LOADS IN 24 SECONDS. Minimum operating system is 4K, and no expansion interface is required to use the unit, which is entirely software controlled. We have had a unit under extensive test for eight weeks now, and cannot get it to do anything wrong! We will be demonstrating this amazing device at the Sydney Computing Show, and, hopefully, taking orders.

The other Big News is that Peter Hartley has developed a real live all-singing-all-dancing TRS-80 JOYSTICK, which is fully addressable from BASIC, and works with any sized Level II machine. Interfacing is a breeze, and the unit is powered from the '80 itself. All you games lovers will be after this one, we know. As we go to press, the price of this unit has not been finalised, but it will certainly be well under $60. We are again hoping to have a sample at the Sydney show, and will be taking pre-release orders on a small deposit.

-=000=-

-=*** NEXT MONTH'S ISSUE ***=-

The June issue of MICRO-80 will contain at least the followlowing...

LUNALERT (LI)	A first class you-land-it-on-the-moon type of game with fascinating graphics, and a program that really applys the laws of physics.
BEETHOVEN'S HELPER (LI)	A must for all you would-be composers out there in Level I land. This aid creates all the chords you'll ever need, at the touch of a keypad.
ALCOTESCHT (LII)	A spoof program that will convince your one-over-the-eight friends to catch a cab and leave their car-keys with you 'till morning.
TAPE BLOCKER (LII)	Set up as an inventory package, this program demonstrates the efficient way to use your cassette for data storage and retrieval.
ACEY DUCEY (LII)	A gamblers delight, brought to your VDU from the pages of MICRO-80. Fun for all who like an occasional flutter - even with matches.
GRAFIX (LII)	In machine language, one of the fastest and most fastinating kalidascope-type programs you'll ever see.
ASSEMBLY LANGUAGE	The next instalment of Eddy Paay's brilliant series - not to be missed.
LEVEL I + II	How to run both Level I and Level II in a single TRS-80! Flick the switch and its dumb again!
CHESS WAR (part 1)	A comparative review of two of the competing '80 compatible Chess Programs - by a Chess Expert. We are certain that this will start a debate that will rage for months.
CASINO GAMES	A gamblers overview of a currently available Casino Games package.
EDUCATIONAL PROGRAMS	The first in a series on the needs of our students, and how to meet them in YOUR programs.

The June issue will also contain all the regular features, such as BETTER BYTES, PROBLEM CORNER, INPUT/OUTPUT (readers' letters), and news of the latest releases from MICRO-80 PRODUCTS. Why run the risk of missing out. Take out a subscription today - there's an order form in every issue.

** BETTER BYTES ** a potpourri of this and that, conducted by Peter Hartley

(Readers' contributions are sought for this column.)

If you've been having problems trying any hardware modifications (except those published in MICRO-80, of course) and get the impression that the published circuits in the Tandy Technical Manual are not about your TRS-80... well, you are probably quite right. It appears that the Mud Map is for the A series circuit boards only, and there have been at least a D series and a G series since, plus the current production - with the new keyboard, which is redesigned around the ROM sockets. I haven't been able to get an opportunity to rip into of one of these latest ones yet, so I don't know what board designation they carry. Some modifications that work fine with one series of boards will just stop another unit from working at all, as I found when trying to retro-fit a Level I ROM into my G board, with provision for running as Level I OR Level II at the flick of a switch. We know how to do this on an A or a D board, and although I finally got mine to run, we will be testing this version for a few weeks before publishing. As you will have seen from the "In Next Month's Issue" column, we are quite satisfied with the A and D board modification at this stage.
The board number is found at the back in the center, when you take off the top cover, on the opposite side from the power supply transistors (the big ones with the heat-sinks).

I'm finally convinced that anything is now possible. While putting last month's issue together, we lost most of the Directory on the floppy that held some 80 hours concentrated typing. Fortunately, I was able to get hold of a copy of TRS-80 Disk and Other Mysteries (reviewed elsewhere in this edition by our resident Disk and Hexadecimal Genius, Eddy), and, with great fear and trepidation, started rebuilding a disk directory byte-by-byte. Obviously it worked, because the April Edition did finally happen. I cannot reccomend that book too highly.

If you live in or around Melbourne, start looking forward to October, because you are going to get to see not one, but TWO computer shows! One for the hobbiest, pure and simple, and the other for the big fellows, but also with a hobbiest section.

I'm still getting odd phone calls and letters about LIFE. Seems that a few of our readers have lost their copies of the magazine, but have the program on tape. Please note that this program was in TWO parts. One in BASIC, and One in m.l. Neither is any use without the other. It was NOT intended to run under Disk BASIC. To use LIFE first power-down, count to 10 and re-power. Answer MEMORY SIZE? with 20479, and then use SYSTEM to load the m.l. tape. When the tape is loaded you can press the BREAK key to get back to BASIC, to load the BASIC routines. Then type RUN, press <ENTER> and away you go.
Ron Sulley, our regular correspondent, has relocated the m.l. program, by altering all the 50H numbers to 70H, and all the 51H numbers to 71H, and thus relocated it to start at 7000H. As I normally use relative jumps wherever possible, this should work well enough (it was written too long ago to remember), but make sure to change the routine addresses in the poke statements at the beginning of the BASIC routine, or you'll crash.

In Level II (NOT DISK) BASIC, if your system crashes, and you get the MEMORY SIZE? bizzo, all may not be lost. Press the RESET button, and you may find that all your efforts are still in there. As I understand it - and I may well be wrong - if you answer MEMORY SIZE? with the <ENTER> key, the '80 will happily trundle its way through all your RAM testing each byte until it finds one that (i) doesn't work, or (ii) isn't there. Now IT knows how much memory you HAVE got, and YOU now know how much program you HAVEN'T got.

We've had a few cases of cassettes sent us coming very unpacked in the post. Please, use a small padded-bag (which you can buy from the Postal People) and keep the outer box at home - this saves postage, and reduces the chance of damage.

Someone out there is about to get a surprise. They ordered some software from MICRO-80 PRPODUCTS, and will have received, instead of what they ordered, our MASTER CASSETTE for the APRIL EDITION. As soon as we know who this lucky person was, we can send them their original order!

On the problems of using a timesharing terminal system - "Long delays make some people positively unpredictable. I know one programmer who put his fist through a terminal keyboard after waiting several minutes for a system to respond" (John Walker, Personal Computing, Nov '78)
Isn't it nice to have your own system ready and waiting all the time?
P.H.

***** ASSEMBLY LANGUAGE PROGRAMMING - PART 4 *****

BY EDWIN R. PAAY.

In the previous article of this series, we discussed Data Movement. This month we will discuss Arithmetic and Logic functions.
When we handle the Logic and Arithmetic functions, take an especial note of the effect these have on the flags in the F register, as this is important for conditional branching which we will discuss later.

* * * LOGIC FUNCTIONS * * *

The Logic functions look like this:

```
10      XOR     A
20      AND     B
30      OR      (HL)
40      OR      12H
50      CPL
60      NEG
```

These instructions show the type of format that the Logic functions might take. The XOR, OR and AND functions all operate on two bytes, the first in the A register and the second can be any register, number, or contents of a memory location. What the computer does is simply line up both bytes one above the other and operate on each vertical pair of bits separately.
Now let us do this for the lines above. Line 10 says: EXCLUSIVE OR the A register with itself. e.g. if the A register contains 75H before execution of line 10 then we have:

```
contents of A register = 75H =  0 1 1 1  0 1 0 1
XOR with itself                 0 1 1 1  0 1 0 1
                          XOR   -----------------
Result will be:               0 0 0 0  0 0 0 0
```

The result will end up in the A register, so after execution of line 10 we have in effect cleared the A register(A will contain zero). This is a convenient way to clear the A register
Line 20 says: AND the contents of the A register with the contents of register B. If before execution register A contains 50H and register B contains 3FH then:

```
contents of A register = 50H =  0 1 0 1  0 0 0 0
contents of B register = 3FH =  0 0 1 1  1 1 1 1
                          AND   -----------------
result will be:          10H =  0 0 0 1  0 0 0 0
```

So the A register will contain 10H after executing line 20.

Some of you might have noticed that I deliberately set out the first four bits in the example above into the format of a truth table. This is to remind us that the "AND" function only returns a " 1 " if both bits to be ANDed are " 1 ". I will do the same with the next example also.
Line 30 says: OR the contents of the A register with the value in the memory location pointed to by the HL register pair. Therefore if HL = 4000H, A=50H and location 4000H = 3FH then:

```
contents of A register  = 50H = 0 1 0 1  0 0 0 0
contents of 4000H       = 3FH = 0 0 1 1  1 1 1 1
                          OR    ----------------
result will be:          7FH    0 1 1 1  1 1 1 1
```

Therefore after execution the A register will contain 7FH.

Line 40 is similar to line 30 except that a direct value is given to OR with the value in the A register. Note that for all the instructions mentioned above (XOR, AND, OR) only the A register is changed after execution. Another very important fact is that the C flag is RESET (Reset means set to ZERO) by all Logic functions. The Z flag however is SET (Set means set to one (1)) if the result in the A register is ZERO and RESET if the result is NOT ZERO. This means that in the examples given so far only line ten would cause the Z flag to be SET.
Now we come to line 50. This says COMPLEMENT the value in the A register This is a bit for bit inversion; in other words, if the A register contains 45H then:

```
contents of A register = 45H =  0 1 0 0  0 1 0 1
                         CPL    ----------------
result will be:          BAH =  1 0 1 1  1 0 1 0
```

After execution of line 50 the A register will contain BAH.
Line 60 is more interesting. This performs the two's complement on the contents of the A register The 2's complement is used to allow both negative and positive numbers to be represented in binary. The idea is that the most significant bit of a binary number is used to signify whether the number is negative or not. If the most significant bit is set the number is negative, and if it is reset it is positive. What this instruction does is to invert the contents of the A register and then increment it by one. The 2's complement of a binary number is defined as that number which, when added to the original number, will result in a sum of zero, ignoring the carry.
Therefore if the A register contains 45H = 0 1 0 0 0 1 0 1
then after execution of line 60 (NEG) ----------------
the result in the A register will be: BBH = 1 0 1 1 1 0 1 1
Now if we were to add the two numbers together:

```
      45H =    0 1 0 0  0 1 0 1
      BBH =    1 0 1 1  1 0 1 1
ADDING        -----------------
RESULT      1  0 0 0 0  0 0 0 0
```

The ninth bit in the result then would be the carry which is ignored (it is lost so to speak), leaving a zero result.
The CPL instruction does not affect the Z or C flags.
The NEG instruction SETs the Z flag if result is zero and RESETs the Z flag if result was not zero. The C flag will be SET if the A register was not ZERO before the NEG operation and RESET otherwise.
The Logic functions are mostly used for MASKING, a term best explained by example. Let us assume that we are writing an A/L pro`ram which uses graphics, and we want to make sure that the addresses used stay within the screen memory (3C00H - 3FFFH). We cduld do this by using masking techniques; for example:

```
100     LD      HL,VIDEO        ;VIDEO = VIDEO ADDRESS
110     LD      A,H             ;GET MSB OF VID. ADDRESS
120     OR      3CH             ;MAKE SURE H REG.>= 3CH
130     AND     3FH              MAKE SURE H REG.<= 3FH
140     LD      H,A             ;LOAD MASKED VALUE IN H REG.
```

These instructions would be part of a larger program. Line 100 loads the HL register pair with an address value labelled VIDEO. Now we want to make sure that this value will never exceed 3FFFH or be smaller than 3C00H. To do this we only have to operate on the most significant byte (MSB) of the video address, because the least significant byte (LSB) hs allowed to vary between 0H and FFH. So in line 110 we load the A register with the MSB of the video address which is contained in the H register, and then in line 120 we do the first masking operation which makes sure that the value will never be lower than 3CH (the OR function will insert bits if it is lower and raise the MSB to 3CH). Anything higher than 3CH will not be affected at all. In line 130 we do the next masking operation to make sure the MSB will not exceed 3FH. If it is lower, this function will not affect the value in the A register at all, but if it is higher than 3FH, ANDing it with 3FH will remove all higher order bits in the byte and convert it to 3FH. Line 140 then will simply load the masked value back into the H register. Try this on a piece of paper for several different values both larger and smaller than the min. and max. values and you will see what I mean. If you would like to see this done in practice then have a look at lines 2640 to 2670 in the source listing of the SET2 program in last months issue.
Now let us cover some Arithmatic operations.

***** ARITHMETIC OPERATIONS *****
Arithmetic operations are available in both 8 bit and 16 bit kinds. The 8 bit ones use the A register as the accumulator and the 16 bit versions use the HL, IX or IY registers for this function. (the IX and IY regis`er's are of limited use as accumulators however)
This is what some of the 8 bit kind of instructions look like:

```
10      ADD     A,B
20      SUB     C
30      ADC     A,0FEH
40      SBC     A,(IX+3)
50      CP      80H
60      INC     D
70      DEC     (HL)
```

Line ten is a simple addition. If register A = 10H and register B = 5H, then after execution A will contain 15H. Line 20 is a simple subtraction with the result ending up in the A register again. Line 30 is an ADD WITH CARRY this is the same as the normal add in line 10 with exception that it also adds the contents of the carry flag to the result. Line 40 is the same as line 20 except that it also subtracts the contents of the carry flag from the result. Instructions of this type, involving a carry, are used for multiple precision Arithmatic involving a number of bytes for each number. The carry is represented by the carry (C) flag.
Line 50 shows a very important function. It is the COMPARE instruction which allows us to compare values so that we can take appropriate action. I will come back to this instruction in more detail later.
Line 60 INCrements the value contained in register D by one (i.e. adds 1 to it), and line 70 does the opposite. DECrementing the value contained in the memory location pointed to by the HL register pair, by one. Of course all types of addressing can be applied to the Arithmatic functions shown above e.g ADD A,(HL) and CP (IY+4) are both legal instructions. Also, I have only given a small sample of possible configurations available.

```
100     ADD     HL,DE
110     ADC     HL,BC
120     SBC     HL,DE
130     ADD     IX,SP
140     INC     DE
150     DEC     BC
```

These are all 16 bit Arithmatic functions and operate in the same manner as the 8 bit equivalents.

* ** FLAGS ****
In all the Arithmatic operations (exepting the 16 bit INC and DEC) the Z flag is set, when the result is zero, and reset otherwise. The C flag will be set if there was a carry from the most significant bit in the result with additions, or a BORROW with subtractions, and reset otherwise. This brings us back to the compare function (CP), this function has exactly the same effect on the flags as a subtraction would have except that the value in the A register is unchanged by it.
let's look at some comPares in detail:

```
5       CP      80H
```

The value in the A register will be compared with 80H in this case. If A = 50H then after execution of line 5 above, the Z flag will be RESET because the value in the A register is not equal to 80H and a s0btraction would not have resulted in zero, and the C flag will be SET because a borrow would have been generated since 80H is larger than the value in the A register. If the A register was 90H, then after executing line five the Z flag would be RESET again, and the C flag would be RESET as well in this case. If A was 80H, then the Z flag will be SET , and the C flag will be RESET.
Now examine the following A/L statement:

```
125     CP      B
```

This compares the value in the accumulator (register A) with the value contained in register B. We can now make the following conclusions regarding the C and Z flags:

The C flag will be set If A is smaller than B (A < B)
The C flag will be reset if A is equal to or larger than B (A >= B)
The Z flag will be set if A equals B (A = B)
The Z flag will be reset if A does not equal B (A <> B)

This makes the C and Z flags the most useful flags to use for conditional branching.

Branching and subroutines will be the next topic of discussion in this series.
For those of you who do not understand Logic Operations at the single bit level some home-work is in order. Information on this subject can be found in almost any book that handles digital electronics and/or assembly language programming.

***** USING M/L SQBROUTINES WITH BASIC *****
At this stage many of you will be able to write BASIC m/l programs (even if you don't think so). There are, however, a few things you must know before you can use m/l routines with BASIC which are not clearly stated in the level 2 manual.
First of all it may be necessary to shift the stack pointer, if the stack is going to be used extensively. As a matter of fact if a m/l program is loaded and run through the "SYSTEM" command provided by BASIC, then the stackpointer MUST be moved. This is due to the fact that BASIC will put the stack right in the middle of its I/O buffer. This can cause problems if ROM calls are made or when trying to return to BASIC. O.K. then. Let us do some supposing... Suppose you have written a machine language program, you have a level 2 16k machine and need to pass a value from basic to your routine and need to pass a value back again. This should just about cover any possible type of m/l subroutine you might come across in the future.

It will be necessary to POKE the entry pointer through basic as explained in the level 2 manual... Suppose that the m/l routine is located a 7F00H. Split this address in two and we get: 7FH and 00H. Convert this to decimal and it becomes 127 and 0. Now we will POKE the entry pointer with the lsb first, which is zero, and then the msb.
A BASIC program to do this might look like this:

```
10 POKE 16526,0:POKE 16527,127:' THIS SETS THE ENTRY POINTER.
20 INPUT" ENTER A NUMBER ";Y :' GET A VALUE FROM THE BOSS.
30 X = USR (Y):' GOTO M/L SUBROUTINE.
40 PRINT" THE ANSWER IS ";X  :' PRINT THE VALUE PASSED BACK.
50 GOTO 20:' DO IT AGAIN.
60 END
```

Whatever your program does is not important at this stage. All we know is that it takes the value from variable Y, then operates on it and places the answer in variable X and returns to the next line in your basic program. Your m/l program will look like this:

```
100             ORG     7F00H   ;START IS AT 7F00H
110  START       CALL    0A7FH   ;GET VALUE FROM VARIABLE "Y"
115  ;
120  ;THE VALUE IS NOW CONTAINED IN THE HL REG. PAIR.
130  ;HERE WOULD BE LOCATED A CERTAIN FUNCTION TO BE USED
110  ;ON THE VALUE IN THE HL REG.
150  ;THE ANSWER MUST BE PLACED IN THE HL REG. PAIR.
160  ;
170              JP      0A9AH   ;PUT ANSWER IN VARIABLE "X"
80              END
```

Line 110 gets the value from variable Y in this case and puts it in the HL register pair. Lines 115 to 160 might contain some sort of function which operates on the value in the HL register pair and leaves or places the result back ia the HL register pair. Line 170 then will put the result in the space allocated to variable "X", and will return to the next line in your basic program. Easy isn't it ? If you don't want to pass a value back to basic then just a RET (return) will do in place of line 170
If your program is one that is used in conjunction with the SYSTEM command as discussed above the the start and end of your program will be different. Let me give an example:

```
100             ORG     7F00H
110  START      LD      (STACK),SP
120             LD      SP,START-4
130 ;
MORE PROGRAM HERE !
990 ;
1000            LD      SP,(STACK)
1010            JP      1A19H
1020 STACK      DEFW    00
1030            END     START
```

This program shows a typical start and end of a m/l program which can be loaded straight from tape and executed. Line 100 again says " the start of the program will be 7F00H. Line 110 say` " load location labeled STACK with the current stack plinter". Line 120 then says " load the SP with an address 4 bytes lower than the start of this program ". This way the stack will never end up in your program area. At the end of the program line 1000 will load the previous stack pointer back (this is not really necessary in this case but illustrates how it can be done.). Line 1010 then will jump back to the BASIC command mode. Line 1020 contains a pseudo op. This is a command to the assembler, and is not part of the a/l program as such. What this says to the assembler is " when you assemble this program leave two bytes (i.e. define a word of two bytes) at this location and fill them with zero's ". This space is labelled STACK and is used to store the stack pointer at the start of the program.

With the information presented so far you should be able to write some simple m/l programs. For those of you who do not have an assembler, hand assembly can be used using the look-up tables provided in the Zilog Assembly Language Programming Manual. This is quite possible. After all, how do you think the first assembler program was written? That's right - by hand!
Now to make sure that you won't forget everything you've learned so far (like you did after the last article) I will give you some homework!

*** HOMEWORK ANYONE?
The problem is this: you have a game program with a rather intricate board patern. Now certain errors could destroy the board or wipe the screen clear. We want a m/l program capable of storing the pattern on the screen, and capable of zapping it back immediately when required. This can be done with the commands handled so far. (hint use the LDIR function)
The reason for using a m/l routine is that BASIC is too SLOW.
Next issue we will discuss how to do this .

-=OOO=-

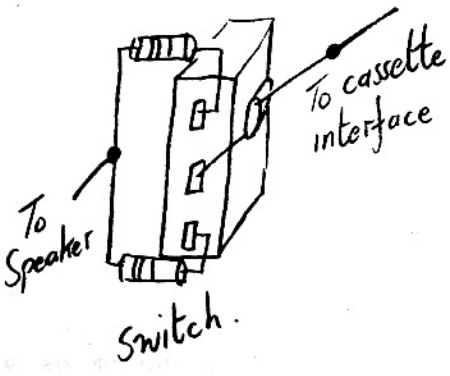
** ON BOARD CASSETTE MONITOR ** by Peter Hartley

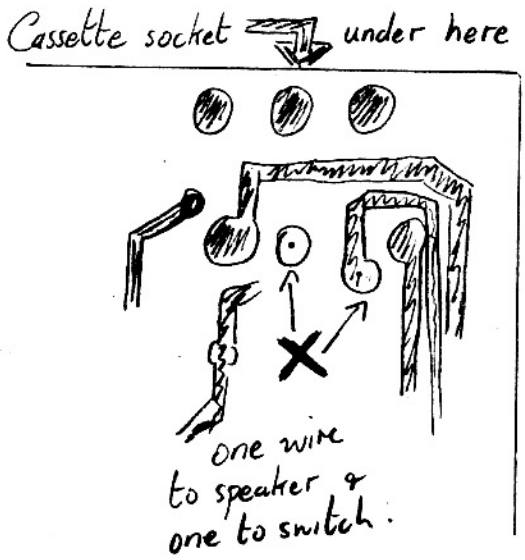
Those of us who are prone to persueing our hobby into the early hours - in spite of pleas of "Come on dear", "You know you've got to be up early in the morning", etc, etc, - are always liable to floods of abuse if we wake the sleeping better(?) half when loading tapes into the '80.

My particular solution was to remove the on-cassette monitor (outlined in the first issue of MICRO 80), and relocate it within the '80, complete with a three-way level control, and audio monitoring of the record function, too!

The whole modification shouldn't set you back more than $6.50. You'll need a small loudspeaker, a one-pole two - way,three position switch, one 100 ohm and one 470 ohm resistor, a low value tantalum capacitor, and a hank of wire.

Open up the '80, and with a good solid adhesive - I used a Sealastic type silicon rubber - glue the speaker, face-down against the vent slots on the underside of the base of the case. Drill a suitable hole in the top of the case (mine is just to the right of the <BREAK> key) for the switch. Solder a wire to each lug of the speaker, and replace the boards in the '80 case.





Solder the capacitor and the two resistors to the switch, as shown in the illustration, and run a length of wire from the capacitor to either one of the contacts X on the main board, behind the cassette take-off socket. One of the speaker wires is then taken to the other X terminal, and soldered in place.

The final connection is to the switch, at the junction of the two resistors, where the other speaker wire is terminated.

Now you can screw it all together again, because you've finished. The two end switch positions will give you loud and soft monitoring, and the central position turns it off.

The monitor will only work when you are recording as long as you really are recording. In the record mode, the cassette deck outputs a monitor signal to the earpiece. This is fed to the monitor speaker. In the play mode the monitor is wired in parallel with the computer input.

** PROBLEM CORNER ** conducted by the MICRO-80 resident hackers.

This month we will devote our efforts to a Victorian Correspondent who has apparently got a bad case of the "GREMLINS".

He has 16K Level II, and writes...
I have made several attempts to make RICOCHET work, including retyping the whole listing and eliminating minor typing errors, but... I am unable to get it past lines 50 or 75. These continually give a F.C.ERROR...

Well now, the real problem here is that RICOCHET was a Level I program, and there is often more to converting these to Level II than just typing them in without the abbreviations.
First off, Level I uses wrap-round graphics. That is to say, although the graphics are arranged 0 to 127 by 0 to 47, you can still set X=190, Y=70, because in Level I, as soon as any graphic value exceeds what is allowed, the computer simply starts subtracting the maximum allowable value from what it has been given until the result is acceptable. So in the example set out, the computer will set X=32 and Y=22 (remember that computers count from ZERO). So, to convert Level I graphics to Level II, you will have to add a routine like this...

```
110 IFX>127 X=X-128:GOTO110
120 IFY>47 Y=Y-48:GOTO120
```

to prevent F.C. ERROR from giving you a thrombosis.
You may also need a similar routine to eliminate any values less than Zero.
Also, with a program like RICOCHET, which simulated the Level II INKEY function by setting certain points on the screen, and then performing IF POINT type operations to see which set points were still there, to determine if there had been an input from the keyboard, and if so which it was, well... there is another catch when you convert to Level II.
Level I gives a 1 and ZERO response to IF POINT type enquiries. Level II gives -1 and ZERO. So, this will need changing also.

Our correspondent continues, "The second problem concerns the Program HANGMAN from issue 2. Once again, after several attempts to get the program to work, including carefully eliminating typing errors, I still find that the program does not give you time to respond with any typed letters, but runs straight through to the end of the program. I have made a temporary "fix" by adding a timing loop between lines 130 and 140, to give me time to think and enter a letter, but it is still apparent that this is not the correct cure..."

Here is a listing of the published version of the appropriate part of HANGMAN...

```
140 PRINT@970,"CHOOSE A LETTER:";
150 A$="":A$=INKEY$:IFA$=""GOTO150
160 IFA$=CHR$(13)THEN150
```

If line 150 was typed in as...

```
150 A$="":A$=INKEY$:IFA$=" "GOTO150
```

this would create exactly the effect that is complained of, just as

```
150 A$="":A$=INKEY$:IFA$="THIS IS NOT A NULL STRING AT ALL"GOTO150
```

would do, too.

To continue..."Another problem now emerges, as the program does not respond to the input letter, until it has completed the next part of the graphics routine, whether this letter is correct or not. Therefore it is impossible to win, even by cheating...."

It seems to us that this could indicate one of two things. With the now added timing loop, it becomes necessary to actually hold down the selected key as the program continues its headlong rush through line 150, otherwise A$ is just a NULL STRING - in which case A$ is wrong, and the player continues to get hung. Our correspondent will know whether A$ is anything else, because if it is, line 155 is...

```
155 PRINT@987,A$
```

which is what happens - but if A$ is a null, you won't see anything happen (because null strings are invisible, even to a TRS-80). Alternatively, the search routine is in lines 190 and 200...

```
190 FORL=1TOLEN(CH$(R))
200 IFMID$(CH$(R),L,1)=A$THEN280ELSENEXT
```

where 280 is "letter accepted", and if anything is wrong in this little area, once again, you get hung a bit more!

We continue..."There is also an apparent problem with the last section, between lines 235 and 320... the program crashes straight through and comes up with a READY."

Buried right in the heart of this offending section, what do we find but...

```
240 A$="":A$=INKEY$:IFA$=""GOTO240
```

and later...

```
250 IF A$<>"Y"THENCLS:END
```

Now this seems to be the same problem as before. The INKEY function scans the keyboard memory to see if a key has been pressed. If it hasn't, the variable receiving the INKEY single letter $tring, gets a null $tring, which in BASIC is written as "" (Two quotation marks with nothing between them). Lines like 240 (above) should loop for ever until either a key is pressed, or there is a power-cut, or your ROM goes bad, because at the end of the line is an expression that says "if nothing was pressed at all, go back, do it again, and have another look at the keyboard memory"

We resume.."The other two problems are from issue 3, and are the machine language programs SET2 and BMON. Both were loaded with the aid of TBUG, and punched out onto the tape O.K. When reloaded, the Hex still matches the magazine listings. As I don't fully understand Hex, I have been most careful to ensure that there are NO errors in my typing.

With SET2, when the program is activated, it fills the screen with garbage, and then, after a few seconds, the screen starts filling from the bottom with letter A's.

When BMON (part1) is activated with "/31641" the copyright message comes up all right, but when I try to access it with the shifted down arrow, once again the computer locks up on me..."

Sir, don't worry... BMON (part1) did this to everyone. Someone, whose name appears right at the top of page one in every edition, left part of the listing out of that issue of the magazine. He still says he's sorry.

As far as SET2 goes however, we can only say - the listing was accurate. Have you set your memory size correctly? Did you use the correct entry and location addresses when you dumped your Hex to tape? Have you tried reloading it with TBUG and making a new tape, taking especial care over these addresses? SET2 really does work as published, and we all suffer from something called HEXADECIMALITIS when entering Eddy Paay's creations. I seem to suffer mild dislexia after as little as 30 minutes of Hex entry. Perhaps a friend would help you "proof read" the in-memory Hex against the published listing.

We resume.."Is it possible to make modifications to the cassette recorder, so as to improve the erasing quality. I find that I have to put tapes through 4 or 5 times to ensure that they are adequately erased, otherwise I have problems with bad data transfer..."
If you are running the cassette off the mains - as is good computing practise - then take it back to your friendly Tandy Manager and spoil his day. One pass should erase all, and the test is to erase something of no value, and then replay it with the volume up full. If you hear ANY residual TRS-80 type conversations, either the erase head is faulty, or the circuit that drives it is crook, or the record head and the erase head are badly aligned. (The most likely cause is that one of the two wires feeding the erase head (the erase head is the white plastic one on the left-hand side) has come off at the head, and a dab of solder should cure all.)

To conclude, our correspondent asks if we can relate any of the Hex or Assembly Language Instructions to Single BASIC Instructions.
Most BASIC instructions convert to many multiple bytes of Hex, but we will add this request to Mr Paay's growing list of "things to do before the end of the week" and endeavour to present something along the lines you request.

We invite readers to direct their curly ones at this column, for our attention. We promise nothing - but the best of our endeavours - often SOLUTIONS are beyond us all!

** KEYBOARD REPEAT ON THE TRS-80? ** by Edwin Paay

The TRS-80 uses a 7 byte buffer when it picks up characters from the keyboard, to check for rollover. This can be used to give a repeat key function to your programs.
The buffer is located from 4036H to 403CH. If we zero these Bytes at intervals, then we only have to hold a key down and it will repeat.
Listed below is a sample program using this principle:

```
10 C=0: I$=""
20 GOSUB 100
30 IF A$=I$ANDA$<>"" THEN C=C+1 ELSE PRINTA$;: I$=A$: C=0: GOTO20
40 IFC=5 THEN GOSUB100: IF I$<>A$ GOTO10 ELSE PRINTA$;:GOTO40
50 GOTO 20
100 A$=INKEY$: FORX=16438 TO 16444: POKEX,0: NEXT: RETURN
```

The keyboard will seem to have a bit of a delay before it initially prints to the screen. This is because BASIC is slow. It does however demonstrate the principle.
Variable C is used as a counter to give a certain delay time before the repeat function is enabled.

** T.R.S.- 80 DISK AND OTHER MYSTERIES ** a review by Eddie Paay.

How many times have you disk users had error messages like : "PARITY ERROR DURING READ" or maybe "GAT READ ERROR" or "HIT READ ERROR" not to speak of "DIRECT STATEMENT IN FILE" appear on the screen ?

Most of you will probably know the helpless feeling that these messages cause. Lets face it, most of us don't even know what HIT or GAT stand for, which makes fixing the error almost impossible.

If you have ever wondered about what causes these errors and how to correct them, you will be delighted to know that the is answers to your questions can now be found in a newly-released book - TRS-80 DISK & OTHER MYSTERIES by H. C. Pennington.

This book explains in detail the organisation and format of the disk, the directory and the hit and gat areas, and what they contain. It also explains how different types of files are stored and how to overcome unknown passwords etc. A large portion of the book also shows how to correct errors on the disk to restore unreadable files or directories. This information is available from nowhere else.

But more than that, it discusses in detail most available DOS operating systems (trsdos, vtos, newdos etc) and utility programs such as RSM-2D, MON3, DEBUG an many other utilities available to the TRS 80 user. It also has details and documentation on NEWDOS+ and its utilities, and this alone should make this book invaluable to the NEWDOS owner, as the documentaion with NEWDOS is relatively poor.

In my opinion it is one of the most useful publications available to the TRS-80 disk user. I suggest to all disk users to get this publication while it is available.

E.P.

***** INPUT / OUTPUT *****
letters to the Editor

From; M.G. Thompson, Moorabin Vic.

I am delighted with your magazine and tapes - excellent value but understanding what I read - I am sorry to say is over my head, and I have to never read articles as often. in a hope that I may absorb more. I take heart in that BASIC was originated for the likes of me, and appreciate that bright people like to show their stuff. So I ask would you publish articles by people not so bright, and identify the pages - with say one star for level 1, two stars for level 2, and three stars if you use m.l., and four stars for a.i. users etc.
Now, in all seriousness, I suggest that you not only number the pages of your magazine, but that you identify the Issue on each page, as they are going to be used as reference material later, when the level of our learning is upgraded (apart from the fact that for one program you may need to refer to several issues).
My particular problem is that some instructions can be mis-interpreted. For example, "TYPE IN ENTER" presumably means the same as "HIT THE WHITE BUTTON".

I feel so miserable after many hours of trying to renumber a program with the tapes received, but cannot manage; I ask can you publish simple instructions which only refer to tapes supplied.
Things are not all bad. I did manage "Anologue Clock" and I like the program so much that I would have liked to have it as the number one program on a fun night tape, with games, and other material, that have been renumbered so that I don't have to fiddle with the tapes when trying to find Tattslotto or Hangman when asked at random.
Furthermore, the "Files" program, I like it! But again could not work out if I could multiply Stock Items by Price, if they were in the data lines, as I could not determine the value assigned to each item of data because of the m.l. sub-routine.

(Thank you for your comments, Mr Thompson. Sorry about going over your head sometimes; we'll try to tone things down a bit, and will very definitely look at imoroving the user instructions - perhaps what we have for the Household Accounts Package elsewhere in this issue is more along the lines of what we should be doing?
As far as the renumbering goes, I must presume that you are using BMON, which did have some teething troubles with the renumbering sequences in its early form (see Input/Output in last month's edition). You shouldn't have any problems any more, just give the routine time to do its thing.
I'm not quite sure why you are wanting to renumber so much material, though. BASIC doesn't care what line numbers are used and in most applications the memory savings from renumbering, say from 1 in steps of 1, are not of great significance.
As for Files, well, you cannot do any math during the input sequences, as the data is all stored in $trings, but you could load your data in to memory with Files, then delete everything except your data, and use math on the data left behind, but remember that the data is still in String form, and you will have to use the VAL function to get real numbers, first.
Your other request has been attended to, already. Hope you like it? Ed.)

May I congratulate you on your magazine which I patiently wait for each month.
I have a TRS-80 Level II which I have converted to 16K, and I have also added the "XRX-III" cassette loading modification, supplied by Tandy. A well worthwhile modification, but one which leads me to a small problem which I seem to have with m.l. programs in your magazine. The problem is that they won't work in my machine. But allow me to explain what has happened.
Firstly I am only a beginner at this game, and when I tried to run the Monitor in Basic (Jan '80), I found that it wouldn't work, of course. Then I loaded the later version, and it worked O.K.
I can now dump a m.l. program to tape, and reload it using the monitor, but the tapes will not load using the System command, although my copy of Microchess loads without trouble. I have very carefully typed in BMON, parts 1 & 2, and checked and rechecked against the listing. The only entry I cannot decipher is 725BH which appears to be 72 or 7E. I have read it as 72?
If I load BMON using the Monitor in Basic, and then type SYSTEM, <ENTER>, /31641, the copyright message appears, but when I press shift and down arrow the machine locks up, showing only a cursor at the top left of the screen - all very confusing when you don't really know what you are doing anyway!
I have tried entering some other m.l. programs with the Monitor in Basic, and using the same method of starting them (SYSTEM, <ENTER>, /entry number) and they all seem to work O.K. so I assume that the Monitor is working correctly. I am now waiting for your next edition in the hope that there was an error in the BMON listing, but if there was not, perhaps you might have some clues as to the nature of my problem?

(I'm sorry to hear that you are having so much strife, but you sound as though you've plenty of perseverance. Tapes made with the Monitor in Basic are not compatible with the SYSTEM system, so to speak, and do have to be loaded and started in the manner you outline. The problem with BMON is very hard to identify. Byte 725BH in that listing was 7E and not 72, but there may yet be other bugs in yOUR listing. Unfortunately, there are literally millions of ways to make a m.l. program cause a lock-up of the type you describe, and I would be only guessing if I said any more than that. By now you will have received the final alterations - in the April edition - and I can only suggest that you and a friend sit down together and proof-read the whole listing. It only needs one character in one byte to be wrong and the whole thing will crash. I'm pleased that, in spite of your troubles, you still "wait at the gate" for MICRO-80 to arrive. We are certainly trying to advance the publication dates towards our target of the first week of the month, but another minor set-back last month disrupted things yet again. As for the XRX-III cassette loader - well, we all use it, and Tandy now fit it to all new units in the factory, so it must be good! Ed.)

***** THIS MONTH'S SOFTWARE *****

** SUB ATTACK ** Level 1/4K By Marlon Binet

Just type it in and RUN. This fun program is another demonstration of level 2 INKEY simulation in level 1.

A battery of guns is drawn along the bottom edge of the screen, and these move in and out - rather like pistons going up and down - in sequence, while nasty enemy subs pop up from out of the water all over the place. All you have to do is hit a key while the gun that's pointing right at the enemy is sticking out. Sounds easy? If you're fast enough to beat this game, you've either got E.S.P. or you're cheating!

** SPACE-DRIVE ** L1/16K

An unusual twist to the "you-land-it" space game. After an automatic take-off, complete with two passengers, you are required to navigate through four meteor showers - each progressively more severe than the last. Demonstrating the level 1 simulation of Big Brother's INKEY function, you have three controls at your command. However, there's a catch. Sabotage has cross-connected the controls, so you may have to learn how to drive all over again with each meteor shower!

Only then can you land. Happy space-driving.

** TRIG/BAS ** (L2) by Robert Sunners

A short, sweet, and very much to the point package from the pen (or nimble keyboard) of Bob Sunners, this will work out all your trig problems. Just tell your '80 what you've got, of any triangle, and this package will fill in all the gaps in the glitch of screen refresh cycle. Keep 'em coming, please, Bob!

** SUPER-SIZZLER ** (L2/8K) by Robert Sunners

The idea of this game is to build a panel-van from the ground up, using components from El Cheepo Spare Parts, at the bottom of the screen. If you are ever lucky enough to get the van built, your '80 will leap in and drive it away!

The catch? Well, to get the parts, you've got to either WIN them or, horror of horrors, PAY for them. To earn your crust, and to win the occasional bonus part, you have to shoot targets through two moving grids. If you happen to hit either of the grids, instead of the targets, you will be billed for repairs to them. If you cannot pay, you may have to sell a part or two back to El Cheepo Spare Parts - and probably lose money in the process. If things get too tough, the '80 will petition fo your bankruptcy, and stop playing.

To find out more - start typing!

** TIC-TAC-TOE ** (L2/4K) by Trevor Henderson

From time-to-time, when the Micro 80 hackers are wading their way through the incoming cassettes, one lobs up that holds a surprise. This is such a program. Our initial reaction was one of "Ho Hum - that old chestnut again" and the cassette went straight to the bottom of the heap. We're delighted to have to admit our mistake.

Trevor's package is a gem, offering FIVE levels of play, ranging from "easy enough for my children to beat it" to "you're lucky to force a draw".

When typing the program, use the shift key with the following...

line	20	"X"	"O"	
line	80	"C"	"S"	"N"
line	200	"X"	"O"	etc
line	740	"N"		

The following commands are all shifted...

C	clear system and restart from game 1.
S	restart the current game.
N	restarts the current game, but changes the player to move first.
X or O	restarts the current game, exchanging noughts and crosses.

L (unshifted) allows you to enter the I.Q. level.
0 identifies the squares.
↓ removes the identification.

Numbers from 1 to 9 input the selected square for play.

At the end of each game the display is held, with the score tally for the series, until the space bar is pressed

A word of warning - level five can generate up to four moves ahead. Players should use care not to fall into some of the less obvious traps that may be generated.

** HOUSEHOLD ACCOUNTS (LII/16K) ** Author Lance Lawes

Written for 16K, cassette and line printer, this program provides a 100 line buffer for storing information. Data may be entered from keyboard or cassette, and may be output to cassette, printer, or VDU. Data may be freely edited, as required, and reports produced by the program include 4 journals, trial balance, and ledger accounts which may be numbered from 1 to 100 inclusive.
Enter the program, as listed, and CSAVE it (or take out a subscription and send for your free software tape as explained in the Editorial).
Let's go through the main MENU, displayed as soon as the program is RUN.
1) Keyboard input Each line of data is input as a string variable. Extensive use is made of the MID$ and VAL functions to manipulate substrings of the main data string. Therefore, the positioning of each of the data fields within the string is important. The video display shows where each field commences. These fields are <date> <reference number> <details> <total> <account number> <debit amount> <credit amount>

Enter <date> as up to six characters and/or spaces
right-arrow key
<reference number> e.g. cheque number, etc
right-arrow key
<details> e.g. payee's name
right-arrow key
<total> e.g. value of cheque
right-arrow key, space
<account number> two parts; prefix CP(cash payments)
CR(cash received)
GJ(general journal)
SJ(sales journal)
followed by the account number.
Thus CP16 would be the account 16 in the cash payments journal.
right-arrow key
<debit> enter debit amount
right-arrow key
<credit> enter credit amount
Now hit <ENTER> and the whole string will be recorded, and the prompt ? will be displayed. To exit this routine, type EXIT and hit <ENTER>.

2) Cassette data input Follow the instructions on the screen. You will need to input the number of data records on file, so remember this whenever you save to cassette.

3) Read memory Displays current contents of buffer

4) Edit Memory Allows you to rewrite any line from present buffer. Use (3), above, to establish correct line number.

5) Save to Cassette Full instructions given on screen

6) Print Journals
7) Print Ledger Balances
9) Ledger Acounts All necessary instructions for these will be displayed on VDU.
8) Lineprinter utility allows direct interface between keyboard and lineprinter.

To use this package the user will need to prepare a chart of accounts... e.g.

BANK ACCOUNT	AC/NO	1
HOUSEHOLD EXPENSES		2
VEHICLE EXPENSES		3
RENT, RATES		4

etc, etc
Suppose that a cheque is now drawn for motor vehicle insurances. The cheque but should be coded with a "3" (vehicle expenses), and the data entry would appear as...

DATE	REFNO	DETAILS	TOTAL	AC/NO	DEBIT	CREDIT
MAR	1001	INSURANCES LTD	123.45	CP03	123.45	

If two policies were paid with the one cheque, the entry might be as...

MAR	1001	INSURANCES LTD	123.45	CP03	100.00	
MAR				CP02		23.45

where multiple dissections apply.

Similarly, cash received might be recorded as...

```
MAR          WAGES GROSS       200.00 CR15          250.00
MAR          WAGES TAX                CR16    50.00
```

thus allowing you to keep track of your income, tax deductions, and any other deductions from source.

You could also keep track of your loan indebtedness by this form of entry...

```
MAR   JNL    VEHICLE EXPENSES          GJ03    100.00
MAR   JNL    BANKCARD                  GJ07          100.00
```

With a little bookkeeping experience, or with much practise, it is quite possible to keep a set of records on the Double Entry System. Suppose we had the following data...

AC/NO	ITEM	ASSIGNED DEBIT	VALUES CREDIT
1	HOUSE, LAND & CONTENTS	50000.00	
2	MOTOR VEHICLE	5000.00	
3	CASH AT BANK	500.00	
4	HOUSE MORTGAGE		20000.00
5	HOUSEHOLDERS CAPITAL		35500.00
	(TRIAL BALANCE TOTALS)	55500.00	55500.00

Whatever breakup of headings that you consider appropriate to your situation can be allocated account numbers, for both out-goings and in-comings. Since the sales-journal will often not be applicable in the normal household situation, that journal could be applied to recording the opening balances each month (which would be obtained from the trial balance at the end of the previous month), and cheque payments, banking deposits, bankcard transactions, etc, could be recorded in the appropriate journals.

```
LINEPRINTER UTILITY
THE LINEPRINTER UTILITY MAY BE USED TO PUT HEADINGS ONTO THE
LINEPRINTER
THE FOLLOWING IS THE INTITAL DATA FED INTO THE BUFFER
TWO ENTRIES ARE STILL TO COME
1) THE TOTAL FOR CASH RECEIVED
2) THE TOTAL FOR CASH PAYMENTS
THIS INFORMATION WILL COME FROM THE PROGRAM ITSELF

**********************************************************************
( 1 ) MAR  CASH AT BANK              SJ01    150.00
( 2 ) MAR  WAGES (YEAR TO DATE)      SJ02          10000.00
( 3 ) MAR  GROUP TAX                 SJ03   2000.00
( 4 ) MAR  SUPERANNUATION            SJ04    500.00
( 5 ) MAR  HOUSEHOLD EXPENSES        SJ05   1000.00
( 6 ) MAR  VEHICLE EXPENSES          SJ06    900.00
( 7 ) MAR  INSURANCE EXPENSES        SJ07    245.00
( 8 ) MAR  INTEREST EXPENSE          SJ08    325.00
( 9 ) MAR  MOTOR VEHICLE             SJ09   5000.00
( 10 ) MAR  HOUSE LAND & CONTENTS    SJ10  50000.00
( 11 ) MAR  HOUSE MORTGAGE           SJ11          20000.00
( 12 ) MAR  BANKCARD                 SJ12            256.00
( 13 ) MAR  HOUSEHOLDERS CAPITAL     SJ13          29764.00
( 14 ) MAR  WAGES GROSS       378.00 CR02            500.00
( 15 ) MAR  WAGES GROUP TAX           CR03    128.00
( 16 ) MAR  WAGES SUPER               CR04     28.00
( 17 ) MAR  WAGES GROSS       378.00 CR02            500.00
( 18 ) MAR  WAGES GROUP TAX           CR03    128.00
( 19 ) MAR  WAGES SUPER               CR04     28.00
( 20 ) MAR  101  CASH          75.00 CP05     65.00
( 21 ) MAR                            CP06     10.00
( 22 ) MAR  102  BANKCARD     256.00 CP12    256.00
( 23 ) MAR  103  INSURANCE LTD 150.00 CP05     50.00
( 24 ) MAR                            CP06    100.00
( 25 ) MAR  104  CASH          75.00 CP05     65.00
( 26 ) MAR                            CP06     10.00
( 27 ) MAR  105  HOUSE LOAN LTD 150.00 CP11    150.00
( 28 ) MAR  JNL  DISCOUNT STORES      GJ05     50.00
( 29 ) MAR  JNL  BANKCARD             GJ12            50.00
( 30 ) MAR  JNL  EL CHEAPO TYRES      GJ06     80.00
( 31 ) MAR  JNL  BANKCARD             GJ12            80.00
( 32 ) MAR  JNL  INTEREST             GJ08     79.00
( 33 ) MAR  JNL  INTEREST             GJ11            79.00
PRINTOUT COMPLETE
```

CASH PAYMENTS JOURNAL FOR (DATE)MAR

DATE	REF NO	DETAILS	TOTAL..	A/C NO	DEBIT...	CREDIT..
MAR	101	CASH	75.00	CP05	65.00	
MAR				CP06	10.00	
MAR	102	BANKCARD	256.00	CP12	256.00	
MAR	103	INSURANCE LTD	150.00	CP05	50.00	
MAR				CP06	100.00	
MAR	104	CASH	75.00	CP05	65.00	
MAR				CP06	10.00	
MAR	105	HOUSE LOAN LTD	150.00	CP11	150.00	
MAR	TOTALS		706.00		706.00	0.00

WE CAN NOW ENTER THE LAST TWO ENTRIES
$706.00 FOR CASH PAYMENTS
$740.00 FOR CASH RECEIVED

CASH RECEIVED JOURNAL FOR (DATE)MAR

DATE	REF NO	DETAILS	TOTAL..	A/C NO	DEBIT...	CREDIT..
MAR		WAGES GROSS	370.00	CR02		500.00
MAR		WAGES GROUP TAX		CR03	100.00	
MAR		WAGES SUPER		CR04	10.00	
MAR		WAGES GROSS	370.00	CR02		500.00
MAR		WAGES GROUP TAX		CR03	100.00	
MAR		WAGES SUPER		CR04	10.00	
MAR	TOTALS		740.00		260.00	1000.00

WE NOW RETURN TO THE MAIN MENU
AND SELECT * 1 * TO KEY IN THE LAST TWO LINES

THE COMPLETED JOURNALS ARE AS FOLLOWS

CASH PAYMENTS JOURNAL FOR (DATE)MAR

DATE	REF NO	DETAILS	TOTAL..	A/C NO	DEBIT...	CREDIT..
MAR	101	CASH	75.00	CP05	65.00	
MAR				CP06	10.00	
MAR	102	BANKCARD	256.00	CP12	256.00	
MAR	103	INSURANCE LTD	150.00	CP05	50.00	
MAR				CP06	100.00	
MAR	104	CASH	75.00	CP05	65.00	
MAR				CP06	10.00	
MAR	105	HOUSE LOAN LTD	150.00	CP11	150.00	
MAR	TOTALS CASH PAYMENTS			CP01		706.00
MAR	TOTALS		706.00		706.00	706.00

CASH RECEIVED JOURNAL FOR (DATE)MAR

DATE	REF NO	DETAILS	TOTAL..	A/C NO	DEBIT...	CREDIT..
MAR		WAGES GROSS	370.00	CR02		500.00
MAR		WAGES GROUP TAX		CR03	100.00	
MAR		WAGES SUPER		CR04	10.00	
MAR		WAGES GROSS	370.00	CR02		500.00
MAR		WAGES GROUP TAX		CR03	100.00	
MAR		WAGES SUPER		CR04	10.00	
MAR	TOTALS CASH RECEIVED			CR01	740.00	
MAR	TOTALS		740.00		1000.00	1000.00

GENERAL JOURNAL FOR (DATE)MAR

DATE	REF NO	DETAILS	TOTAL..	A/C NO	DEBIT...	CREDIT..
MAR	JNL	DISCOUNT STORES		GJ05	50.00	
MAR	JNL	BANKCARD		GJ12		50.00
MAR	JNL	EL CHEAPO TYRES		GJ06	80.00	
MAR	JNL	BANKCARD		GJ12		30.00
MAR	JNL	INTEREST		GJ08	79.00	
MAR	JNL	INTEREST		GJ11		79.00
MAR	TOTALS				0.00	209.00 209.00

~~SALES JOURNAL~~ OPENING BALANCES FOR (DATE)MAR

DATE	REF NO	DETAILS	TOTAL...	A/C NO	DEBIT...	CREDIT..
MAR		CASH AT BANK		SJ01	150.00	
MAR		WAGES (YEAR TO DATE)		SJ02		10000.00
MAR		GROUP TAX		SJ03	2000.00	
MAR		SUPERANNUATION		SJ04	500.00	
MAR		HOUSEHOLD EXPENSES		SJ05	1600.00	
MAR		VEHICLE EXPENSES		SJ06	800.00	
MAR		INSURANCE EXPENSES		SJ07	245.00	
MAR		INTEREST EXPENSE		SJ08	325.00	
MAR		MOTOR VEHICLE		SJ09	5000.00	
MAR		HOUSE LAND & CONTENTS		SJ10	50000.00	
MAR		HOUSE MORTGAGE		SJ11		20000.00
MAR		BANKCARD		SJ12		256.00
MAR		HOUSEHOLDERS CAPITAL		SJ13		29764.00
MAR	TOTALS				60020.00	60020.00

DALEK CHASE (L2/16K & up) $15.00 + 50c p&p.
An absorbing game in which you save Dr Who from persuing Daleks. The better you play, the harder it gets (and vica versa). Infinate range of play levels from HARD to UTTERLY IMPOSSIBLE, and comes complete with messages of praise and mediocre insults. (If you are really bad enough at this game, it actually refuses to play any more!) ENTHRALLING FUN FOR ALL.

UBOAT $15.00 + 50c p&p
Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedos, plenty of targets, working fuel and battery meters, helpful Mothership for high-sea reprovisioning, and even has emergency radio for that terrible moment when the depth-charges put your crew at risk. Requires Level II/16K.

** Coupon duplicate appears on inside back page *

* Coupon prices are inclusive of post & packing *

To: MICRO-80, P.O. Box 213, Goodwood, S.A. 5034.

Please rush me the items checked below.

.....	12 month subscription to MICRO-80 and my free software catalogue.	$24.00
.....	12 month subscription to MICRO-80 and the cassette edition, plus my free software catalogue.	$60.00
.....	The latest issue of MICRO-80.	$ 2.50
.....	Touchtype.	$20.45
.....	RPN Calculator	$25.45
.....	BMON	$20.45
.....	Dalek Chase	$15.50
.....	Learning Level II	$18.15
.....	BOX of 10 * SCOTCH C10 Tapes	$24.00
.....	Box of 10 * SCOTCH C30 Tapes	$28.00
.....	Lower Case Mod.	$ 6.00
.....	Me Think it MICRO-mazing v.1	$15.50
.....	Me Think it MICRO-mazing v.2	$15.50
.....	UBOAT	$15.50
.....	Space Invaders	$ 9.00
.....	TRS-80 ROM Handbook .	$26.00

----------TOTAL ENCLOSED $.....

Please debit my Bankcard $.....

Number.....Expires.....

Name.....

Address.....

.....

.....Post Code.....

Bankcard

MICRO-80 PRODUCTS MIC

MICRO-80 PRODUCTS MIC

MICRO-80 PRODUCTS MIC

SCOTCH BRAND COMPUTIN
Available in two size
the TOPS. Individual
rigid box, they come
C10 (five mins per si
$24.00 incl p&p
C30 (fifteen mins per
$28.00 incl p&p
These tapes come
specification demande
and are not recommend

BMON - THE ULTMATE
(L2/16K & up) - (
Basic) $15.00 + 50c
Our own personnel
first loading this a
program into high me
Basic programs on the
loading; Tells you
program just loaded;
through; Merges t
renumbering of the s
clashes of line numb
though you did type N
while you work on a
time!); Lists all the
Makes SYSTEM tapes; L
the list goes on and
32K, and 48K versi
afford NOT to have BM
** N.B. The price
$19.95 from June 1st.
firstly to do justice
secondly because the t
all three popular me
our manufacturing cost
today!

ME THINK IT MICRO' MAZI
two three game cassett
Highly animated and
master of TRS-80 gra
brings you THIEF (stea
TANK (a two player b
SHOOTOUT (Mexican styl
offers KNEVIEL (as i
INDY 500 (the ulti
(blowing up unfortun
the place). All are r
II/4K

80 PRODUCTS MICRO-80 PRODUCTS

80 PRODUCTS MICRO-80 PRODUCTS

80 PRODUCTS MICRO-80 PRODUCTS

APES
these blank cassettes are
packaged in an extra strong,
packs of ten.

de)

ight up to the highest
f SCOTCH for digital tapes,
for audio use.

H MEMORY BASIC MONITOR -
: suitable for use with Disk
).
use to write BASIC without
ng machine language utility
y! BMON Renumbers; Displays
reen while they are still
e memory locations of the
s you stop a load part-way
programs, with automatic
nd so as to prevent any
; Recovers your program even
Makes one program invisible
ond (saves hours of cassette
riables used in the program;
you Edit memory directly...
Cassette comes with 16K,
ready to load. Can anyone

MON will be increased to
e have decided to do this,
o the author, Eddy Paay, and
now contains versions for
y sizes, which has increased
So get your order off

$15.00 + 50c p&p each
rtaining games from that
cs, Charlie Bartlett. No 1
n old man's hoard of loot),
le of speed and skill), and
igh-noon for two). No 2
torcycle, ramp and buses),
road-race) and WARSHIP
little submarines all over
time simulations for Level

RPN CALCULATOR (L2/16K & L2/32K) $24.95 +50c p&p.
Give your computer the power of a $650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed, whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette NOW comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32k programmable version to run in 16K. Whether for everyday or occasional use, this package will prove invaluable, and turn your '80 into a very powerful instrument.

TOUCHTYPE (L2/4K) $19.95 +50c p&p.
An interactive, 22 lesson typing course which uses the computer's keyboard and screen to teach you to type rapidly and accurately, and a massive cassette data dump to control your progress. The computer checks for accuracy, and sets timed excercises to check your progress. If you have to look at each key before you press it, or only use two fingers, then this program, plus a little perseverence, will do some amazing things to your typing speed.

LOWER CASE MOD. KIT $5.50 + 50c p&p
Complete in every regard, comes with prime quality makers speciication low power memory chip, switch, wire, and very detailed instructions. Easy to install, and a lot less expensive that the TANDY modification. Stop your '80 from holding out on you, and get your kit TODAY.

LEARNING LEVEL II (book) $16.95 + $1.20 p&p
By the author of the acclaimed Tandy Level I handbook, this reveals the secrets of Level II in the same inimitable style. Bright, breezy, easy to read and understand throughout, it is not only a book in itself, but contains pages to cut out and paste into the Level I manual, converting that to Level II, too! Just reprinted, and specially air-freighted in from the U.S. for YOU, to help you get the most from your '80.

SPACE INVADERS $8.50 + 50c p&p
Much improved version of this arcade favourite, with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II, with 4K and 16K versions on this cassette.

A PRINTOUT OF LEDGER ACCOUNTS IS AS FOLLOWS

DATE	REF NO	DETAILS	TOTAL	A/C NO	DEBIT	CREDIT
MAR		CASH AT BANK		SJ01	150.00	
MAR		TOTALS CASH RECEIVED		CR01	740.00	
MAR		TOTALS CASH PAYMENTS		CP01		706.00
TOTALS.....					890.00	706.00
BALANCE OF ACCOUNT.....					184.00	

DATE	REF NO	DETAILS	TOTAL	A/C NO	DEBIT	CREDIT
MAR	WAGES	YEAR END		SJ02		10000.00
MAR		WAGES DEPOT	370.00	CR02		500.00
MAR		WAGES GROSS	370.00	CR02		500.00
TOTALS.....					0.00	11,000.00
BALANCE OF ACCOUNT.....					11,000.00-	

DATE	REF NO	DETAILS	TOTAL	A/C NO	DEBIT	CREDIT
MAR		GROUP TAX		SJ03	2000.00	
MAR		WAGES GROUP TAX		CR03	100.00	
MAR		WAGES GROUP TAX		CR03	100.00	
TOTALS.....					2,200.00	0.00
BALANCE OF ACCOUNT.....					2,200.00	

DATE	REF NO	DETAILS	TOTAL	A/C NO	DEBIT	CREDIT
MAR		BANKCARD		SJ12		256.00
MAR	102	BANKCARD	256.00	CP12	256.00	
MAR	JNL	BANKCARD		GJ12		50.00
MAR	JNL	BANKCARD		GJ12		80.00
TOTALS.....					256.00	386.00
BALANCE OF ACCOUNT.....					130.00-	

A TRIAL BALANCE FOR THE END OF THE MONTH IS AS FOLLOWS

DISSECTION TOTALS	
1	184.00
2	11,000.00-
3	2,200.00
4	500.00
5	1,220.00
6	1,000.00
7	245.00
8	404.00
9	5,000.00
10	50,000.00
11	19,929.00-
12	130.00-
13	29,764.00-
TOTAL	0.00

END OF RUN

THESE CLOSING BALANCES WILL BECOME THE OPENING BALANCES FOR NEXT MONTH

This program requires 15K, so does not leave room to expand the buffer in a 16K machine. Those of you with larger memories, and busier bank-accounts or small business applications, will need to edit line 10 (clear 7000), and all the program lines with FOR/NEXT loops (180, 300, 240, 370, 600, 690, 1100, 1380)

** CONNECTX and CONNECTA ** (L2) by Robert Sunners.

Two little routines for plotting lines from (X1,Y1) to (X2,Y2), without using m.l. routines like SET2. Both are simple, self explanatory, and as you can have them both up-and-running in ten minutes, we won't tell you any more now!

```
10  REM SUB ATTACK BY MARLON BINET 14/2/80
20  CLS:P."WELCOME TO SUB ATTACK"
30  IN."DO YOU WANT THE RULES (1=YES,2=NO)";Q
40  IFQ=1CLS:GOSUB30000
50  CLS:GOS.2000
60  S=0:L=30:P.A.1000;
70  P.A.911;L;:P.A.975;S;
80  P.A.70;"HERE WE GO . . .";
90  F.I=1TO3000:N.I
100  P.A.70;"                    ";
110  GOS.3000
120  X=RND(70)+24:Y=(RND(09)+1)*3+1:GOS.1000
130  GOS.4000
140  IFL=0T.31000
150  G.120
1000  REM SUB TO DRAW SUB AT X,Y
1010  F.I=XTOX+10:S.(I,Y):N.I:F.I=X+2TOX+8:S.(I,Y+1):N.I
1020  F.I=X+1TOX+4:S.(I,Y-1):S.(I+6,Y-1):N.I
1030  S.(X+4,Y-2):S.(X+4,Y-3):RET.
2000  REM SUB TO DRAW GUNS
2005  P.A.896;
2010  F.I=0TO127:S.(I,40):S.(I,0):N.I
2020  F.I=0TO40:S.(0,I):S.(127,I):N.I
2030  P.A.896;"SHOTS LEFT . .";
2040   P.A.960;" SUBS HIT . . ";
2050  F.I=36TO86S.10:GOS.2500:N.I:RET.
2500  F.J=ITOI+4:S.(J,38):S.(J,39):N.J
2510  S.(I+2,37):S.(I+2,36):RET.
3000  REM SUB TO SET CURSOR ETC
3010  P.A.1000;:F.I=79TO82:S.(I,46):N.I:RET.
4000  F.I=1TO3:F.J=38TO88S.10
4010  S.(J,35):IFP.(80,46)=0T.GOS.5000:RET.
4015  F.K=1TO50:N.K:R.(J,35)
4020  N.J:N.I:C=I.(Y/3)*64+X/2-1:P.A.C;"          ";
4025  L=L-1
4030  P.A.C-64;"          ";:GOS.3000:RET.
5000  IFP.(79,46)=0T.O=-1:G.5030
5010  IFP.(82,46)=0T.O=0:G.5030
5020  O=1
5030  F.I=35TOY-3S.-1:S.(J,I):R.(J,I):J=J+O:N.I
5040  J=J-3*O:L=L-1:P.A.911;L;
5050  R=I.(Y/3-1)*64+X/2
5060  IFA.(J-X-5)>6P.A.R;"       ";:P.A.R+64;"       ";:GOS.3000:RET.
5070  F.I=1TO500:N.I:P.A.R;"BOOM!";
5080  F.I=1TO1000:N.I:P.A.R;"       ";:P.A.R+64;"       ";:S=S+1
5090  P.A.975;S;:GOS.3000:RET.
30000  P."WELCOME TO SUB ATTACK"
30010  P."YOUR MISSION IS TO DESTROY AS MANY OF SUBS"
30020  P."AS POSSIBLE WITH A LIMITED NUMBER OF SHOTS."
30030  P."THE SUBS WILL SURFACE AT RANDOM PLACES ON THE BOARD"
30040  P."AND WILL REMAIN ONLY FOR A LIMITED TIME."
30050  P."IF YOU ALLOW THE SUB TO ESCAPE WITHOUT BEING"
30060  P."SHOT AT YOU FORFEIT A SHOT."
30070  P."TO ACHEIVE ALL THIS YOU COMMAND A NUMBER OF GUNS"
30080  P."SITUATED AT THE BOTTOM OF THE SCREEN THE BARRELS"
30090  P."OF WHICH MOVE IN AND OUT IN SEQUENCE.THE GUN WHICH"
30100  P."FIRES AT ANY ONE TIME IS THE ONE WITH IT'S BARREL"
30110  P."OUT.YOU MAY MAKE THE GUNS FIRE IN A CHOICE"
```

```
30120  P."OF THREE DIRECTIONS BY PRESSING THE APPROPRIATE KEY."
30130  P."SPACE BAR FIRES FORWARD,] FIRES TO THE LEFT AND ↑"
30140  P."FIRES TOWARD THE RIGHT."
30150  IN."TYPE 'ENTER' TO CONTINUE";A$:CLS
30160  P."YOUR SCORE AND REMAINING SHOTS ARE DISPLAYED"
30170  P."AT THE BOTTOM OF THE SCREEN."
30180  P."THAT IS ALL YOU HAVE TO KNOW"
30190  P."            G O O D L U C K  ! !"
30200  IN."TYPE 'ENTER' WHEN READY";A$:CLS:RET.
31000  CLS:P."YOU SCORED A ";S;"OUT OF A POSSIBLE 30"
31010  INPUT"CARE TO PLAY AGAIN (1=YES,2=NO)";Q
31020  IFQ=1T.50
```

SPACE DRIVE (L1)

```
10 CLS:Y=1:N=0
20 PRINT"DO YOU REQUIRE INSTRUCTIONS  (Y/N)  ";:INPUTA
30 IFA=0GOTO60
40 IFA=1GOTO6000
50 GOTO20
60 CLS:Y=26:FORX=51TO62:SET(X,Y):NEXTX:X=47
70 FORY=27TO30:FORX=XTOX+3:SET(X,Y):NEXTX:X=X-8:NEXTY
80 X=63:FORY=27TO30:FORX=XTOX+3:SET(X,Y):NEXTX:NEXTY
90 SET(35,31):SET(78,31)
100 FORX=35TO78:SET(X,32):NEXTX
110 FORY=33TO37:SET(40,Y):SET(41,Y):SET(56,Y):SET(57,Y)
120 SET(73,Y):SET(74,Y):NEXTY
130 FORY=38TO40:FORX=14TO99:SET(X,Y):NEXTX:NEXTY
190 PRINTAT330;"BOARDING SOON"
200 FORE=1TO1000:NEXTE:PRINTAT330
210 RESET(35,31):GOSUB1300:RESET(35,32):GOSUB1300
215 E=14
220 Y=32:FORA=0TO10STEP2:SET(35-A,Y):SET(34-A,Y)
230 FORZ=1TO50:NEXTZ:Y=Y+1:NEXTA
240 Y=37:FORX=14TO21STEP2:SET(X,Y):GOSUB1400:RESET(X,Y):NEXTX
250 Y=36:X=24:FORA=0TO5:SET(X,Y):GOSUB1400
260 RESET(X,Y):X=X+2:Y=Y-1:NEXTA
270 X=36:Y=31:FORX=XTOX+ESTEP2:SET(X,Y):GOSUB1400
280 RESET(X,Y):NEXTX:E=E-5
290 SET(X,Y):IFE<=3GOTO310
300 GOTO240
310 Y=37:X=24:FORA=0TO10STEP2:RESET(X,Y):RESET(X+1,Y)
320 FORZ=1TO50:NEXTZ:Y=Y-1:X=X+2:NEXTA:GOSUB1300
330 SET(35,32):GOSUB1300:SET(35,31):GOSUB1300
370 PRINTAT276;"STAND BY FOR TAKE OFF"
380 FORX=10TO0STEP-1
390 FORY=1TO300:NEXTY
400 PRINTAT349;X:NEXTX
405 PRINTAT276:PRINTAT349
410 X=40:FORY=37TO33STEP-1:RESET(X+34,Y):RESET(X+33,Y)
420 RESET(X+17,Y):RESET(X+16,Y):RESET(X+1,Y):RESET(X,Y)
430 NEXTY
440 PRINTAT768:GOSUB1300:GOSUB1400
450 PRINTAT832:GOSUB1300:GOSUB1400
460 FORA=1TO11:PRINTAT960:FORQ=1TO20:NEXTQ:NEXTA:Q=20
500 CLS:Q=Q+20
510 IFQ=120GOTO3000
520 A=RND(63):B=RND(63)+63:C=RND(63)+127:D=RND(63)+191
530 E=RND(63)+255:F=RND(63)+319:X=3
```

```
540 FORZ=1TORND(18)
550 PRINTAT0:PRINTATA;"*":A=A-2:IFA<=3THENA=RND(63)
560 GOSUB1500
570 PRINTAT64:PRINTATB;"*":B=B-1:IFB<=67THENB=RND(63)+63
580 GOSUB1500
590 PRINTAT128:PRINTATC;"*":C=C-1:IFC<=131THENC=RND(63)+127
600 GOSUB1500
610 PRINTAT192:PRINTATD;"*":D=D-2:IFD<=195THEND=RND(63)+191
620 GOSUB1500
630 PRINTAT256:PRINTATE;"*":E=E-1:IFE<=259THENE=RND(63)+255
640 GOSUB1500
650 PRINTAT320:PRINTATF;"*":F=F-1:IFF<=323THENF=RND(63)+319
660 GOSUB1500
670 NEXTZ
700 CLS:FORZ=1TO75:PRINTAT135;"PREPARE FOR METEOR SHOWER"
705 PRINTAT135:NEXTZ
710 SET(X,Y):SET(X+1,Y):SET(X+2,Y)
720 FORA=5TO120:SET(A,10):SET(A,40):NEXTA
730 FORA=10TO40:SET(5,A):SET(120,A):NEXTA
740 FORZ=1TOQ
750 A=RND(114)+5
760 B=RND(29)+10
770 SET(A,B)
780 RESET(X,Y+1):RESET(X+1,Y+1):RESET(X+2,Y+1):RESET(X+3,Y+1)
790 RESET(X+4,Y+1):RESET(X+3,Y):RESET(X+4,Y):RESET(X,Y-1)
800 RESET(X+1,Y-1):RESET(X+2,Y-1):RESET(X+3,Y-1):RESET(X+4,Y-1)
820 NEXTZ
830 GOTO1900
900 A$=ENTER
910 B$=]--
920 PRINTAT896:PRINTAT896;"PRESS ";A$;" TO GO DOWN.     ";B$;" TO ";
930 PRINT"REVERSE.     OR  P  TO GO UP.":GOSUB1600
940 IF(POINT(2,4)=0)*(POINT(6,4)=1)*(POINT(8,4)=1)GOTO1000
950 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=1)GOTO1100
960 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=0)GOTO1200
970 RESET(X-1,Y):RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y)
980 X=X+2:SET(X,Y):SET(X+1,Y):SET(X+2,Y)
990 GOSUB1700:GOTO940
1000 PRINTAT896:PRINTAT896;"PRESS ";A$;" TO GO DOWN.    ";B$;" TO ";
1010 PRINT"GO FORWARD.    P  TO GO UP":GOSUB1600
1020 IF(POINT(2,4)=0)*(POINT(6,4)=1)*(POINT(8,4)=1)GOTO920
1030 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=1)GOTO1100
1040 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=0)GOTO1200
1050 RESET(X-1,Y):RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y)
1060 X=X-2:SET(X,Y):SET(X+1,Y):SET(X+2,Y)
1070 GOSUB1700:GOTO1020
1100 PRINTAT896:PRINTAT896;"PRESS ";A$;" TO GO DOWN.   ";B$;" TO ";
1110 PRINT"REVERSE.    OR  P  TO GO FORWARD":GOSUB1600
1120 IF(POINT(2,4)=0)*(POINT(6,4)=1)*(POINT(8,4)=1)GOTO1000
1130 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=1)GOTO920
1140 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=0)GOTO1200
1150 RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y)
1155 Y=Y-1:SET(X,Y):SET(X+1,Y):SET(X+2,Y)
1160 GOSUB1700:GOTO1120
1200 PRINTAT896:PRINTAT896;"PRESS ";A$;" TO GO FORWARD.   ";B$;" TO ";
1210 PRINT"REVERSE.     P TO GO UP.":GOSUB1600
```

```
1220 IF(POINT(2,4)=0)*(POINT(6,4)=1)*(POINT(8,4)=1)GOTO1000
1230 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=1)GOTO1100
1240 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=0)GOTO920
1250 RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y)
1260 Y=Y+1:SET(X,Y):SET(X+1,Y):SET(X+2,Y)
1270 GOSUB1700:GOTO1220
1300 FORZ=1TO150:NEXTZ:RETURN
1400 FORZ=1TO80:NEXTZ:RETURN
1500 RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y):X=X+1
1510 SET(X,Y):SET(X+1,Y):SET(X+2,Y):RETURN
1600 PRINTAT66;:SET(2,4):SET(6,4):SET(8,4):RETURN
1700 IF(POINT(X+3,Y)=1)+(POINT(X-1,Y)=1)+(POINT(X,Y-1)=1)GOTO1800
1710 IF(POINT(X+1,Y-1)=1)+(POINT(X+2,Y-1)=1)+(POINT(X,Y+1)=1)GOTO1800
1720 IF(POINT(X+1,Y+1)=1)+(POINT(X+2,Y+1)=1)GOTO1800
1740 IF(X=1)+(X=123)GOTO500
1750 RETURN
1800 I=INT(Y/3)*64+(X/2)
1810 FORO=1TO10
1820 PRINTATINT(I)-1;"]***↑";
1830 GOSUB1300
1840 PRINTATINT(I)-1;"CRASH";
1850 GOSUB1300
1860 N.O
1870 CLS:GOTO5750
1900 IF(X>=63)*(Y>25)GOTO1950
1910 IF(X>=63)*(Y<25)GOTO1970
1920 IF(X<63)*(Y>25)GOTO1990
1930 FORK=113TO120:FORJ=37TO39:RESET(K,J):NEXTJ:NEXTK
1940 GOTO900
1950 FORK=5TO12:FORJ=11TO13:RESET(K,J):NEXTJ:NEXTK
1960 GOTO900
1970 FORK=5TO12:FORJ=37TO39:RESET(K,J):NEXTJ:NEXTK
1980 GOTO900
1990 FORK=113TO120:FORJ=11TO13:RESET(K,J):NEXTJ:NEXTK
2000 GOTO900
3000 CLS:Y=1:N=0
3010 PRINT"CONGRATULATIONS YOU HAVE MADE IT THIS FAR NOW ALL YOU HAVE"
3020 PRINT"TO DO IS LAND YOUR SPACE SHIP."
3030 PRINT:INPUT"DO YOU REQUIRE LANDING INSTRUCTIONS  (Y/N)";A
3040 IFA=1GOTO7000
3050 IFA=0GOTO3100
3060 GOTO3030
3100 CLS:F=RND(1000)+3500:S=RND(200)+400
3110 PRINTAT825;"\";:PRINTAT953;"[";:GOSUB1600
3120 FORX=116TO125:FORY=35TO45:SET(X,Y):NEXTY:NEXTX
3130 FORX=110TO113:FORY=35TO38:SET(X,Y):NEXTY:NEXTX
3140 FORX=110TO113:FORY=42TO45:SET(X,Y):NEXTY:NEXTX
3150 SET(114,35):SET(115,35):SET(114,45):SET(115,45)
3155 FORX=116TO118:FORY=39TO41:RESET(X,Y):NEXTY:NEXTX
3160 FORX=1TO109:FORY=42TO45:SET(X,Y):NEXTY:NEXTX
3200 Y=9
3210 X=10:Y=Y+1
3220 GOSUB4000:GOSUB1600
3230 GOSUB4000:PRINTAT66;:F=F-1:IFF<=0GOTO4220
3240 RESET(X-1,Y):RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y)
3250 X=X+2:SET(X,Y):SET(X+1,Y):SET(X+2,Y)
```

```
3260 IF(POINT(2,4)=0)*(POINT(6,4)=1)*(POINT(8,4)=1)GOTO3370
3270 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=1)GOTO3420
3280 IF(POINT(2,4)=1)*(POINT(6,4)=0)*(POINT(8,4)=0)GOTO3470
3290 IF(POINT(2,4)=0)*(POINT(6,4)=0)*(POINT(8,4)=0)GOTO3220
3300 IF(POINT(2,4)=0)*(POINT(6,4)=0)*(POINT(8,4)=1)GOTO3220
3310 IF(POINT(X+3,Y)=1)+(POINT(X+2,Y+1)=1)GOTO3500
3320 IF(POINT(X+2,Y-1)=1)+(POINT(X,Y+1)=1)GOTO3500
3330 IFPOINT(119,44)=0GOTO5000
3340 IFX>=121GOTO3360
3350 GOTO3230
3360 RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y):GOTO3210
3370 SET(2,4):W=RND(20):S=S-W:IFS<=0GOTO4200
3380 IFW<5GOTO3220
3390 F=F-(W*5):IFF<=0GOTO4220
3400 Y=Y+1
3410 RESET(X,Y-1):RESET(X+1,Y-1):RESET(X+2,Y-1):GOTO3230
3420 SET(6,4):W=RND(20)
3430 IFW<5GOTO3220
3440 F=F-(W*5):IFF<=0GOTO4220
3450 Y=Y-1
3460 RESET(X,Y+1):RESET(X+1,Y+1):RESET(X+2,Y+1):GOTO3220
3470 SET(6,4):SET(8,4):S=S-RND(2):IFS<=0GOTO4200
3480 GOTO3220
3500 IFX>120THENX=120
3510 IFX<5THENX=5
3520 GOTO1800
4000 PRINTAT10;"FUEL SUPPLY ";F;" LITRES.     SPEED ";S;" M.P.H."
4010 RETURN
4200 IF(S=0)+(S=-1)GOTO5000
4210 S=0:PRINTAT340;"YOU'RE SPEED IS NOW AT ZERO !!!!!":GOTO4230
4220 F=0:PRINTAT340;"YOU ARE NOW OUT OF FUEL !!!!!":
4230 RESET(X,Y):RESET(X+1,Y):RESET(X+2,Y)
4235 Y=Y+1:SET(X,Y):SET(X+1,Y):SET(X+2,Y)
4240 GOSUB4000:IFY=42GOTO4260
4250 GOTO4230
4260 Y=39:GOTO1800
5000 IF(F>=0)*(S>=-1)*(Y=40)*((X=112)+(X=113)+(X=114)+(X=115))GOTO5500
5010 IFPOINT(119,44)=1GOTO4210
5020 CLS:PRINTAT340;"YOU EITHER CRASHED INTO THE HANGER"
5030 PRINTAT406;"OR ELSE YOU CHEATED !!!!!":GOTO5750
5500 CLS:PRINTAT340;"CONGRATULATIONS YOU MADE IT HOME":GOTO5750
5750 PRINT:PRINT"TO PLAY AGAIN ENTER 1 IF NOT ENTER 2";:INPUTA
5760 IFA=1GOTO60
5770 CLS:PRINTAT340;"SEE YOU NEXT TIME"
5780 FORZ=1TO2000:NEXTZ:CLS:END
6000 CLS:PRINT"                              SPACE DRIVE"
6010 PRINT:PRINT"THE IDEA OF THIS GAME IS THAT YOU ARE THE PILOT OF A"
6020 PRINT"SPACE SHIP, AND YOU WILL BE REQUIRED TO STEER YOUR SPACE"
6030 PRINT"SHIP THROUGH FOUR METEOR SHOWERS AND THEN BRING IT INTO"
6040 PRINT"LAND IN THE SPACE CENTRE."
6050 PRINT:PRINT"AT THE START OF THE GAME YOUR SPACE SHIP WILL BE"
6060 PRINT"DRAWN.  AFTER A FEW SECONDS YOU AND YOUR CREW OF TWO WILL"
6070 PRINT"BOARD THE SPACE SHIP AND HAVING BOARDED, THE SPACE SHIP"
6080 PRINT"WILL TAKE OFF.  THE SPACE SHIP IS ON AUTOMATIC PILOT ALL"
6090 PRINT"THE TIME EXCEPT FOR DURING METEOR SHOWERS AND WHEN YOU"
6100 PRINT"WANT TO LAND, AT THESE TIMES YOU WILL BE REQUIRED TO"
6110 PRINT"STEER THE SHIP MANUALLY TO SAFETY."
6120 GOS.8000
```

```
6130 CLS:PRINTAT10;"INSTRUCTIONS FOR METEOR SHOWERS !"
6140 PRINT:PRINT"AFTER YOUR SHIP HAS TAKEN OFF, IT WILL RE-APPEAR ON"
6145 PRINT"THE LEFT OF THE SCREEN FLYING THROUGH SPACE ON AUTOMATIC
6150 PRINT"PILOT.  AFTER A WHILE A METEOR SHOWER WARNING WILL COME ON"
6160 PRINT"THE SCREEN.  YOUR SPACE SHIP WILL BE PUT INTO A CONFINED"
6170 PRINT"AREA, AND METEORS WILL START APPEARING.  A GAP WILL APPEAR"
6180 PRINT"IN THE AREA YOU ARE CONFINED IN AND IT IS YOUR JOB TO STEER"
6190 PRINT"THE SPACE SHIP PAST THE METEORS AND THROUGH THE GAP."
6200 GOSUB8000
6210 CLS:PRINT"THE SPACE SHIP WILL START TO MOVE FORWARD AND WILL"
6220 PRINT"CONTINUE TO DO SO  UNTIL YOU CHANGE DIRECTION. INSTRUCTIONS"
6230 PRINT"ARE GIVEN AT THE BOTTOM OF THE SCREEN ON HOW TO CHANGE"
6240 PRINT"DIRECTION.   ONCE YOU HAVE CHANGED DIRECTION THE SHIP WILL"
6250 PRINT"CONTINUE IN THAT DIRECTION UNTIL YOU CHANGE IT AGAIN."
6260 PRINT:PRINT"NOW A WORD OF WARNING."
6270 PRINT:PRINT"SOMEONE BACK AT THE BASE WAS JEALOUS BECAUSE HE COULD"
6280 PRINT"NOT GO IN PLACE OF YOU AND HE HAS TAMPERED WITH THE STEERING
6290 PRINT"AS A RESULT THE STEERING WILL ALTER AS YOU CHANGE DIRECTION"
6300 PRINT"THE INSTRUCTIONS AT THE BOTTOM OF THE SCREEN ARE ALWAYS"
6310 PRINT"CORRECT"
6320 GOSUB8000
6330 CLS:PRINT"SHOULD YOU HIT A METEOR OR HIT THE BOUNDRY OF YOUR"
6340 PRINT"CONFINED AREA YOU WILL CRASH AND YOU AND YOUR CREW WILL"
6350 PRINT"BE KILLED.      EACH METEOR SHOWER WILL GET WORSE THAN"
6360 PRINT"THE PREVIOUS ONE.  THE FIRST ONE IS ONLY A SMALL ONE AND"
6370 PRINT"WILL GIVE YOU TIME TO CHECK YOUR STEERING  WITHOUT MUCH"
6380 PRINT"CHANCE OF HITTING ANYTHING."
6390 PRINT:PRINT"ONCE YOU HAVE STEERED YOUR WAY THROUGH FOUR METEOR"
6400 PRINT"SHOWERS YOU WILL BE READY TO BRING YOUR SPACE SHIP IN TO"
6410 PRINT"LAND AT THE SPACE STATION.   INSTRUCTIONS FOR LANDING"
6420 PRINT"WILL BE GIVEN AFTER YOU HAVE SUCCESSFULLY GOT THROUGH"
6430 PRINT"THE FOUR METEOR SHOWERS."
6440 PRINT:INPUT"NOW PRESS ENTER TO START THE GAME ";A$
6450 GOTO60
7000 CLS:PRINTAT20;"INSTRUCTIONS FOR LANDING."
7010 PRINT:PRINT"THERE ARE FOUR THINGS ON THE SCREEN YOU HAVE TO WATCH"
7020 PRINT"1. AT THE BOTTOM OF THE SCREEN IS THE LANDING STRIP WITH"
7030 PRINT"THE HANGER AT THE END.  YOUR SPACE SHIP  MUST COME TO REST"
7040 PRINT"BETWEEN THE TWO ARROWS, WITHOUT TOUCHING THE TOP, BOTTOM"
7050 PRINT"OR END OF THE HANGER."
7060 PRINT"2. ON THE TOP LEFT OF THE SCREEN IS YOUR FUEL SUPPLY"
7070 PRINT"3. ON THE TOP RIGHT OF THE SCREEN IS YOUR SPEED INDICATOR"
7080 PRINT"4. THE POSITION OF YOUR SPACE SHIP."
7090 GOSUB8000
7100 CLS:PRINT"THE SPACE SHIP WILL COMMENCE ITS LANDING APPROACH FROM"
7110 PRINT"THE TOP LEFT OF THE SCREEN AND WILL KEEP MOVING FORWARD."
7120 PRINT"AS THE SPEED DECREASES  THE SHIP WILL GRADUALLY LOSE HEIGHT.
7130 PRINT"YOU MAY DO AS MANY ORBITS OF THE SPACE STATION AS YOU WANT"
7140 PRINT"PROVIDED YOU DON'T RUN OUT OF FUEL."
7145 GOSUB8000
7150 CLS:PRINT"YOUR CONTROLS FOR LANDING ARE:-"
7160 PRINT"1    PRESS THE ]-- KEY AND YOUR SPEED WILL BE DECREASSED"
7170 PRINT"BY UP TO 20 M.P.H."
7180 PRINT"2    PRESS ENTER AND YOUR SPEED WILL BE DECREASED BY EITHER"
7190 PRINT"ONE OR TWO M.P.H."
7200 PRINT"3    PRESS P AND YOUR SPACE SHIP WILL GAIN HEIGHT."
7205 GOSUB8000
```

```
7210 CLS:PRINT"IF YOU APPROACH THE HANGER TOO FAST YOU CAN FLY OVER"
7220 PRINT"THE TOP AND COME IN ON THE NEXT ORBIT.  IF YOU ARE TOO LOW"
7230 PRINT"YOU CAN PRESS THE P KEY AND GAIN MORE HEIGHT FOR YOUR SHIP."
7240 PRINT:PRINT"YOUR SPACE SHIP MUST LAND BETWEEN THE TWO ARROWS BUT"
7250 PRINT"NOT TOUCHING ANYTHING.  WHEN THE SPACE SHIP LANDS THERE,"
7260 PRINT"PRESS ENTER.   YOUR SPEED AFTER YOU PRESS ENTER AT THIS"
7270 PRINT"STAGE MUST BE ZERO. IF IT IS ABOVE ZERO YOU WILL CONTINUE"
7280 PRINT"ON AND CRASH INTO THE END OF THE HANGER.  IF YOUR SPEED IS"
7290 PRINT"BELOW ZERO YOUR SPACE SHIP WILL STOP AND CRASH TO EARTH."
7300 PRINT"LIKEWISE IF YOU RUN OUT OF FUEL AT ANY STAGE YOUR SPACE SHIP
7310 PRINT"WILL FALL TO EARTH. ALSO IF IT TOUCHES THE RUNWAY AT ANY"
7320 PRINT"STAGE IT WILL CRASH."
7330 PRINT:PRINT"NOW WHEN READY TO START YOUR LANDING APPROACH";
7340 INPUT" PRESS ENTER";A$
7350 GOTO3100
8000 INPUT"PRESS ENTER TO CONTINUE.";A$:RETURN
```

** LEVEL II SOFTWARE LISTINGS **

These have been processed, prior to printing, to improve the general comprehensibility of each program line. A single space has been inserted after each colon, to open up the listing and make it clearer to those of our readers not yet fully experienced in Level II. There is no need to include these additional spaces when you enter these programs on your own '80. Most print statements will commence on a new line. The MICRO-80 printer still refuses, in spite of much encouragement from us all, to print a downward pointing arrow, which we would like to insert where ever the original source contained a line-feed. Instead you will find EITHER a square bracket, OR an upward arrow. This will help you replicate the original source program. Ed.

```
 30 REM ** TRIG/BAS .... FINDING SIDES OR ANGLES OF TRIANGLES **

 40 REM ** ROBERT A SUNNERS, 26 SUNCROFT ST,
 50 REM ** MT. GRAVATT, BRISBANE 4122 07 349 2598
100 CLS: A=0: H=0: O=0: ANG=0: DEG#=0: MIN#=0: SEC#=0: SD=0:
    S$=" ": T$=" ": S2=0
110 FORX=30TO95: SET(X,6): NEXT
120 FORX=94TO95: FORY=6TO17: SET(X,Y): NEXT: NEXT
130 Y=6: FORX=30TO95: SET(X,Y): Y=Y+11/65: NEXT
140 FORX=88TO93: SET(X,9): NEXT
150 FORX=88TO89: FORY=7TO8: SET(X,Y): NEXT: NEXT
160 PRINT@95,"A";: PRINT@79,
    "ANGLE";
170 PRINT@305,"O";: PRINT@351,"H";: PRINT@177,"90";
180 PRINT@512," "
220 GOSUB1000
221 IFO>0ANDANG>0THENGOTO230
222 IFA>0ANDANG>0THENGOTO360
223 IFH>0ANDANG>0THENGOTO310
225 IFO>0ANDA>0THENGOTO230
226 IFO>0ANDH>0THENGOTO310
227 IFA>0ANDH>0THENGOTO360
```

```
 230 'TANGENT
 240 IF O=0THENO=A*TAN(ANG*.0174533): PRINT@305,INT(O*10000+.5)/10000;:
     GOTO265
 250 IFA=0THENA=O/TAN(ANG*.0174533): PRINT@95,INT(A*10000+.5)/10000;:
     GOTO265
 255 IFO>0ANDA>0ANDANG>0THENGOTO310
 260 IFANG=0THENGOSUB6000: ANG=C: GOSUB6500: PRINT@79,D1;D3;D6;
 265 IFH=0THENGOTO221
 270 PRINT@960," ";: INPUT: GOTO100
 300 GOSUB1000 'SINE
 310 IFO=0THENO=H*SIN(ANG*.0174533): PRINT@305,INT(O*10000+.5)/10000;:
     GOTO335
 320 IFH=0THENH=O/SIN(ANG*.0174533): PRINT@351,INT(H*10000+.5)/10000;:
     GOTO335
 325 IFO>0ANDH>0ANDANG>0THENGOTO360
 330 IFANG=0THENS=O/H: GOSUB5000: ANG=Y: GOSUB6500: PRINT@79,D1;D3;D6;
 335 IFA=0THENGOTO221
 340 PRINT@960," ";: INPUT: GOTO100
 350 GOSUB1000 'COSINE
 360 IFA=0THENA=H*COS(ANG*.0174533): PRINT@95,INT(A*10000+.5)/10000;:
     GOTO385
 370 IFH=0THENH=A/COS(ANG*.0174533): PRINT@351,INT(H*10000+.5)/10000;:
     GOTO385
 375 IFA>0ANDH>0ANDANG>0THENGOTO310
 380 IFANG=0THENS=A/H: GOSUB5500: ANG=Y: GOSUB6500: PRINT@79,D1;D3;D6;
 385 IFO=0THENGOTO221
 390 PRINT@960," ";: INPUT: GOTO100
1000 PRINT,"ENTER ANGLE       -       DEGREES      ";: INPUTDEG#
1005 IFDEG#>=90THENPRINT@640,"NOT TRIANGLE";: GOTO1000
1010 PRINT,"                              MINUTES      ";: INPUTMIN#
1015 IFMIN#>59THENPRINT@640,"TRY AGAIN";: GOTO1000
1020 PRINT,"                              SECONDS      ";: INPUTSEC#
1025 IFSEC#>59THENPRINT@640,"TRY AGAIN";: GOTO1000
1030 PRINT,"ENTER KNOWN SIDE NAME, LENGTH        ";: INPUTS$,SD
1040 ANG=INT((DEG#+MIN#/60+SEC#/3600)*10000+.5)/10000
1045 IFANG=0 PRINT,"ENTER OTHER KNOWN SIDE, LENGTH       ";: INPUTT$,S2
1050 IFS$="A"PRINT@95,SD;: A=SD: GOTO1080
1060 IFS$="H"PRINT@351,SD;: H=SD: GOTO1080
1070 IFS$="O"PRINT@305,SD;: O=SD
1080 GOSUB6500: PRINT@79,D1;D3;D6;
1090 IFT$="A"PRINT@95,S2;: A=S2: GOTO1120
1100 IFT$="H"PRINT@351,S2;: H=S2: GOTO1120
1110 IFT$="O"PRINT@305,S2;: O=S2
1120 RETURN
4990 END
5000 ' SINE
5010 X=S: IFABS(S)<=.707107THEN5060
5020 X=1-S*S: IFX<0THENPRINTS;
     "IS OUT OF RANGE": STOP
5030 W=X/2: Z=0
5040 Y=(X/W-W)/2: IF(Y=0)+(Y=Z)THENX=W: GOTO5060
5050 W=W+Y: Z=Y: GOTO5040
5060 Y=X+X[3/6+X[5*.075+X[7*4.464286E-2
5070 W=Y+X[9*3.038194E-2
5080 IFABS(S)>.707107THENW=1.570796-W
5090 Y=W*57.29578: RETURN
5490 END
5500 ' COSINE
5510 GOSUB5000: Y=90-Y: W=1.570796-W: RETURN
```

```
5590 END
6000 ' TANGENT
6010 X=O/A: IFX<0THENT=-1
6020 IFX=0THENT=0
6030 IFX>0THENT=1
6040 X=ABS(X): C=0
6050 IFX>1THENC=1: X=1/X
6060 A=X[2
6070 B=((2.86623E-3*A-1.61657E-2)*A+4.29096E-2)*A
6080 B=((((B-7.5289E-2)*A+.106563)*A-.142089)*A+.199936)*A
6090 A=((B-.333332)*A+1)*X
6100 IFC=1THENA=1.570796-A
5110 A=T*A: C=A*57.29578: RETURN
6490 END
6500 ' ANGLE CONVERSION
6510 D1=INT(ANG): D2=ANG-D1 'DEGREES
6520 D3=INT(D2*60): D4=D2*60 'MINUTES
6530 D5=D4-D3: D6=INT(D5*60) 'SECONDS
6540 RETURN
```

```
 5 'AUTHOR  T. N. HENDERSON, 3 REECE STREET, LUINA, TAS, 7351

 20 DEFINTQ: Z=3: X=1: Y=1: Q$="x": A$="o": P$=
    "COMPUTER
 30 GOSUB40: GOTO30
 40 CLS: PRINT@5,
    "GAME "X,"     T I C   T A C   T O E","   IQ =
    ": PRINT@296,
    "YOU HAVE  "Q$: PRINT@424,P$
    " HAS  "A$: PRINT@85,STRING$(21,CHR$(130))
 50 A=0: B=0: C=0: D=0: E=0: F=0: G=0: H=0: I=0: FORM=0TO35:
    SET(M,13): SET(M,19): NEXT: FORN=8TO24: SET(11,N): SET(25,N):
    NEXT: IFY=1THEN80
 60 R$=A$: T=1: RANDOM: V=RND(9): ONVGOTO610,620,630,640,600,650,660,670,
    680
 80 R$=Q$: T=11: PRINT@859,
    "YOUR TURN": W$=INKEY$: IFW$="c"THENRUNELSEIFW$="s"THENRETURNELSEIFW$=
    "n"THEN730ELSEIFW$="0"THEN760ELSEIFW$="["THEN242
100 IFW$="1"ANDA=0THEN610ELSEIFW$="2"ANDB=0THEN620ELSEIFW$="3"ANDC=
    0THEN630ELSEIFW$="4"ANDD=0THEN640
140 IFW$="5"ANDE=0THEN600ELSEIFW$="6"ANDF=0THEN650ELSEIFW$="7"ANDG=
    0THEN660ELSEIFW$="8"ANDH=0THEN670ELSEIFW$="9"ANDI=0THEN680
200 IFW$="x"ORW$="o"Q$=W$: IFQ$="o"THENA$="x": RETURNELSEA$="o":
    RETURN
210 IFW$="L"PRINT@859,;: INPUT
    " IQ = ";Z: IFZ<1ORZ>5Z=3
220 PRINT@56,Z;: GOTO80
240 IFA+B+C=66ORF-E+D=55ORI-H+G=88ORG-D-A=22ORH-E-B=11ORI-F+C=66ORI-E-A=
    33ORG-E+C=55PRINT@849,"GOOD MOVE!!     Y O U   W I N":  J=J+1:
    GOTO730
242 PRINT@128," ";: PRINT@134," ";: PRINT@141," ";: PRINT@320," ";:
    PRINT@326," ";: PRINT@333," ";: PRINT@512," ";: PRINT@518," ";:
    PRINT@525," ";: IFW$="["THEN80ELSE710
250 R$=A$: T=1: PRINT@853,P$"'S": IFC+F=90RA+E=60RH-G=1IFI=0THEN680
260 IFA+C=40RH-E=3IFB=0THEN620
270 IFA+B=30RI-F=30RG-E=2IFC=0THEN630
280 IFA+G=80RF-E=2IFD=0THEN640
290 IFH-B=60RF-D=20RI-A=80RG-C=4IFE=0THEN600
```

```
300 IFI-C=60RD+E=9IFF=0THEN650
310 IFA+D=50RC+E=80RI-H=1IFG=0THEN660
320 IFB+E=70RI-G=2IFH=0THEN670
340 FORQ=2TO25*Z: SET(Q+2,33): RESET(Q-2,33): NEXT: IFB+C=50RG-D=30RI-E=
    40RC-B=110RG-D=330RI-E=44IFA=0THEN610
350 IFA+C=440RH-E=33IFB=0THEN620
360 IFA+B=330RI-F=330RG-E=22IFC=0THEN630
370 IFA+G=880RF-E=11IFD=0THEN640
380 IFH-B=660RF-D=220RI-A=880RG-C=44IFE=0THEN600
390 IFI-C=660RD+E=99IFF=0THEN650
400 IFA+D=550RC+E=880RI-H=11IFG=0THEN660
410 IFB+E=770RI-G=22IFH=0THEN670
420 IFC+F=990RA+E=660RH-G=11IFI=0THEN680
430 ONZGOTO60,600,530,510,450
450 IFH-F-E-B=150RH-D-E-B=370RH-D-F-E=330RH-D-F-B=360RH-E-I-B=220RE=55ANDD-
    F-B=360RI-D-G-A=470RI-D-E-A=490RH-A-I-E=630RI-E-A=430RG-F-E=6IFC=
    0THEN630
490 IFF-B-H-E=310RE-B-H-A=240RF-A-I-C=430RI-B-E-A=710RF-A-I-E=410RH-F-E=
    170RH-C-E=50IFG=0THEN660
510 IFD-B-H-E=90RD-B-F-E=110RD-B-H-A=130RD-C-G-E=90RG-B-E-C=470RH-A-E=
    72IFI=0THEN680
520 IFG-F-E-C=30RH-F-D-B=160RH-F-I-B=110RH-C-I-G=390RG-F-I+C=50RC-D=290RG-
    B=750RB-H=140RH-G=810RC-I=240RG-I=68IFA=0THEN610
525 IFI-C=960RD-F=380RF-D=620RG-A=76IFB=0THEN620
530 IFG+E=620RG+A=180RI-B=970RA-F=50RA-I=2IFC=0THEN630
535 IFG-C-E=39IFB=0THEN620
540 IFI-G=920RH-G=81IFD=0THEN640
570 IFA-C=80RE-C=520RA-H=30RI-D=950RI-A=98IFG=0THEN660
590 IFA-E=60RE-A=540RH-B=860RC-A=320RC-H=250RG-F=710RC-G=260RG-C=74IFI=
    0THEN680
600 IFE=0PRINT@329,R$;: E=5*T: GOTO700
610 IFA=0PRINT@194,R$;: A=1*T: GOTO700
620 IFB=0PRINT@201,R$;: B=2*T: GOTO700
630 IFC=0PRINT@207,R$;: C=3*T: GOTO700
640 IFD=0PRINT@322,R$;: D=4*T: GOTO700
650 IFF=0PRINT@335,R$;: F=6*T: GOTO700
660 IFG=0PRINT@450,R$;: G=7*T: GOTO700
670 IFH=0PRINT@457,R$;: H=8*T: GOTO700
680 IFI=0PRINT@463,R$;: I=9*T: GOTO700
690 IFZ=1ANDI>0IFA=0ORB=0ORC=0ORD=0ORE=0ORF=0ORG=0ORH=0THEN600
700 IFT=11THEN240ELSEPRINT@704,CHR$(31): IFA+B+C=60RF-E+D=50RI-H+G=80RG-D-
    A=20RH-E-B=10RI-F+C=60RI-E-A=30RG-E+C=5PRINT@850,P$" WINS      TRY AGAIN
    ":  K=K+1: GOTO730
710 IFA=0ORB=0ORC=0ORD=0ORE=0ORF=0ORG=0ORH=0ORI=0IFT=11THEN250ELSEIFT=
    1THEN80
720 L=L+1: PRINT@855,
    "N O   W I N N E R
730 PRINT@616,"DRAWN GAMES  "L: PRINT@714,
    "YOU HAVE WON   "J,"   "P$" HAS WON
    "K: PRINT@983,
    "GAME "X" FINISHED
    ";: IFY=1Y=2ELSEY=1
740 IFW$="n"RETURN
750 W$=INKEY$: IFW$=" "THENX=X+1: RETURNELSE750
760 PRINT@128,"1";: PRINT@134,"2";: PRINT@141,"3";: PRINT@320,"4";:
    PRINT@326,"5";: PRINT@333,"6";: PRINT@512,"7";: PRINT@518,"8";:
    PRINT@525,"9";: GOTO80
```

```
          HOUSEHOLD BUDGET↑
            AUTHOR L J LAWES, 21 RODNEY ST, LINDUM, 4178, QLD↑
            COPYRIGHT (C) 1980 MICRO-80, P.O.BOX 213, GOODWOOD,↑
            S.A. 5034.  TELEPHONE (08) - 71 9683
 10 CLS: CLEAR 7000: DEFDBLL: DEFSTRA: DEFINTI,J,X: DIMA(101):
    DIMLB(101): DIMLT(101)
 20 F1$="DATE  REF NO  DETAILS           TOTAL...A/C NO  DEBIT...CREDIT..'
 40 F3$="######.##": F4$=
    "#,###,###.##-"
 50 PRINT"                      * * * M E N U * * *"
 60 PRINT"* 1 *  KEYBOARD INPUT","* 6 *  PRINT JOURNALS
    "
 70 PRINT"* 2 *  CASSETTE INPUT","* 7 *  PRINT LEDGER BALANCES
    "
 80 PRINT"* 3 *  READ MEMORY","* 8 *  LINEPRINTER UTILITY
    "
 90 PRINT"* 4 *  EDIT MEMORY","* 9 *  LEDGER ACCOUNTS
    "
100 PRINT"* 5 *  SAVE ON CASSETTE"
150 X$="": X$=INKEY$: X=VAL(X$): IFX$=""THEN150
160 ON X GOTO 170,230,280,340,360,500,1010,1260,1300
170 CLS: PRINT
    "  DATE  REF NO  DETAILS         TOTAL... A/C NO DEBIT...CREDIT.
171 PRINT"  !.....!.......!.............!,---.. !.....!,---..!,---.."
180 FORI=1TO100
190 IFA(I)<>""THEN210
200 A="": INPUT A: IF LEN(A)>62THEN205
203 IFA="EXIT"THEN220
204 GOTO206
205 PRINT"ERROR...ENTRY EXCEEDS MAXIMUM LENGTH": GOTO200
206 A(I)=A
210 NEXT
215 I=I-4
220 CLS: PRINT
    "END OF BATCH...MEMORY CONTAINS ";I;" RECORDS": II=I: GOTO50
230 INPUT"HOW MANY RECORDS ARE THERE ON CASSETTE ";II
235 X$="": INPUT
    "PRESS <ENTER> WHEN READY TO LOAD DATA ";X$
240 FOR I=1TOIISTEP4
250 INPUT#-1,A(I),A(I+1),A(I+2),A(I+3)
255 PRINTA(I): PRINTA(I+1): PRINTA(I+2): PRINTA(I+3)
260 NEXT
270 CLS: PRINT
    "INPUT FROM CASSETTE COMPLETE": GOTO50
280 CLS
290 PRINT"CONTENTS OF MEMORY"
300 FOR I=1TO100
310 PRINT "<";I;">": PRINT A(I): IFA(I)=""THEN330
320 IFI=80RI=160RI=240RI=320RI=400RI=480RI=560RI=640RI=720RI=800RI=880RI=
    96THENFORT%=1TO250:  NEXTT%
325 NEXTI
330 PRINT"CONTENTS OF MEMORY READ": INPUT
    "PRESS <ENTER> TO CONTINUE "; X: CLS: GOTO50
340 CLS: PRINT
    "MEMORY EDIT": INPUT
    "WHICH LINE NO DO YOU REQUIRE";I
350 PRINTSTRING$(10,92);"  CURRENT LINE UNDER REVIEW  ";STRING$(10,92):
    PRINT"  ";A(I): PRINT
    "IF AMENDMENT REQUIRED RE-TYPE LINE...OTHERWISE HIT <ENTER>↑
    HIT <ENTER> TO VIEW THE NEXT LINE IN THE BUFFER↑
    TYPE & ENTER 'EXIT' TO RETURN TO MENU":   INPUT A(I)
```

```
351 Z=0: INPUT Z$: Z=VAL(Z$)
352 IFZ$="EXIT"THENZ$="": GOTO50
353 IFZ=0THEN I=I+1: GOTO350
354 IFZ<>0THEN I=Z: GOTO350
360 PRINT"SAVING ON CASSETTE": INPUT
    "PREPARE CASSETTE...HIT <ENTER>";X
365 PRINT"DATA NOW BEING RECORDED ON CASSETTE...PLEASE WAIT"
370 FOR I=1TO100STEP4
380 PRINT#-1,A(I),A(I+1),A(I+2),A(I+3)
385 PRINTA(I): PRINTA(I+1): PRINTA(I+2): PRINTA(I+3)
390 IF A(I)=""THEN405
400 NEXT
405 I=I-4
410 PRINT"RECORDING COMPLETE...THERE ARE ";I;" RECORDS ON TAPE":
    GOTO50
490 INPUT"<ENTER>";GO
500 PRINT: PRINT: PRINT
    "                    * * * JOURNALS AVAILABLE * * *"
510 PRINT"* 1 *  PRINTOUT OF MEMORY"
520 PRINT"* 2 *  CP  CASH PAYMENTS JOURNAL"
530 PRINT"* 3 *  CR  CASH RECEIVED JOURNAL"
540 PRINT"* 4 *  GJ  GENERAL JOURNAL"
550 PRINT"* 5 *  SJ  SALES JOURNAL"
560 PRINT"* 6 *  RETURN TO MAIN MENU"
570 X$="": X$=INKEY$: X=VAL(X$): IFX$=""THEN570
580 ON X GOTO 590,660,660,660,660,50
590 CLS: PRINT
    "PRINTOUT OF MEMORY"
600 FOR I=1TO100
610 IFA(I)=""THEN640
620 PRINT"<";I;">": PRINT A(I)
630 LPRINT"<";I;">";"  ";A(I)
640 NEXT
650 PRINT"PRINTOUT COMPLETE": LPRINT
    "PRINTOUT COMPLETE": GOTO500
660 INPUT"IS THE LINEPRINTER REQUIRED (Y/N)";P$
665 IFP$<>"Y" AND P$<>"N" THEN660
670 INPUT"WHICH DATE DO YOU REQUIRE        ";E$
680 GOSUB900
690 FOR I=1TO100
700 IF A(I)=""THEN780
710 IF E$<>LEFT$(A(I),6) AND KA$<>MID$(A(I),40,2) THEN780
720 PRINTA(I): IFP$="N"THEN750
730 IFP$="Y"THEN LPRINT A(I)
750 DA$=MID$(A(I),31,8): DA#=VAL(DA$): DG#=DG#+DA#
760 DR$=MID$(A(I),47,8): DR#=VAL(DR$): DT#=DT#+DR#
770 CR$=MID$(A(I),55,8): CR#=VAL(CR$): CT#=CT#+CR#: BL#=BL#+DR#-CR#
780 NEXT
790  PRINT E$;"   TOTALS";: PRINTTAB(31)USINGF4$;DG#;: PRINTTAB(47)
    USINGF3$;DT#;CT#
800 IFP$="Y"THEN LPRINT E$;
    "   TOTALS";: LPRINTTAB(31)USINGF4$;DG#;: LPRINTTAB(47)USINGF3$;DT#;C
810 DG#=0: DT#=0: CT#=0: GOTO490
820 INPUT"IS THE PRINTER REQUIRED (Y/N) ";P$
830 IFP$<>"Y" AND P$<>"N" THEN820 ELSE RETURN
900 ON X-1 GOTO910,940,960,980
910 CLS: PRINT
    "CASH PAYMENTS JOURNAL      FOR (DATE)";E$: PRINT F1$
920 IF P$="Y"THEN LPRINT
    "CASH PAYMENTS JOURNAL     FOR (DATE)";E$: LPRINT F1$
925 KA$="CP": GOTO1000
940 CLS: PRINT
    "CASH RECEIVED JOURNAL     FOR (DATE)";E$: PRINT F1$
```

```
 950 IFP$="Y"THEN LPRINT
     "CASH RECEIVED JOURNAL     FOR (DATE)";E$: LPRINT F1$
 955 KA$="CR": GOTO1000
 960 CLS: PRINT
     "GENERAL JOURNAL           FOR (DATE)";E$: PRINTF1$
 970 IFP$="Y"THEN LPRINT
     "GENERAL JOURNAL           FOR (DATE)";E$: LPRINTF1$
 975 KA$="GJ": GOTO1000
 980 CLS: PRINT
     "SALES JOURNAL             FOR (DATE)";E$: PRINTF1$
 990 IFP$="Y"THEN LPRINT
     "SALES JOURNAL             FOR (DATE)";E$: LPRINTF1$
 995 KA$="SJ"
1000 RETURN
1010 CLS
1020 PRINT"* * * LEDGER BALANCE OPTIONS * * *"
1030 PRINT"* 1 * DISSECTION SUMMARY THIS BATCH"
1040 PRINT"* 2 * RETURN TO MAIN MENU"
1070 GOSUB820
1080 X$="": X$=INKEY$: X=VAL(X$): IF X$=""THEN1080
1090 ON X GOTO 1100,50
1100 FOR J=1TO100: LB(J)=0: NEXTJ
1110 FOR I=1TO100
1120 J=VAL(MID$(A(I),42,2)): L1=VAL(MID$(A(I),47,8)): L2=VAL(MID$(A(I),55,8)
     ):  L=L1-L2: LB(J)=LB(J)+L: LT(J)=LT(J)+LB(J): L=0: L1=0:
     L2=0
1140 NEXTI
1150 PRINT"DISSECTION TOTALS": IFP$="Y"THENLPRINT
     "DISSECTION TOTALS"
1160 FOR J=1TO100
1170 IF LB(J)=0THEN1200
1180 PRINTJ,: PRINTTAB(16)USINGF4$;LB(J): L=L+LB(J)
1190 IFP$="Y"THENLPRINTCHR$(15)J,,: LPRINTTAB(48)USINGF4$;LB(J):
     LPRINTCHR$(14)
1200 NEXTJ
1210 PRINT"TOTAL";: PRINTTAB(16)USINGF4$;L
1220 IFP$="Y"THENLPRINT
     "TOTAL",: LPRINTTAB(48)USINGF4$;L
1230 L=0: PRINT
     "END OF RUN"
1240 IFP$="Y"THENLPRINT
     "END OF RUN"
1250 INPUT"PRESS ENTER TO  CONTINUE ";X: GOTO1010
1260 CLS: PRINT: PRINT
     "LINEPRINTER UTILITY↑
     (TYPE HEADINGS OR NOTES AS REQUIRED - TYPE 'EXIT' TO RETURN TO↑
     THE MAIN MENU)"
1270 M$="": INPUTM$: IFM$=
     "EXIT"THEN50
1280 PRINTM$: LPRINTM$
1290 GOTO1270
1300 PRINT"          * * * LEDGER ACCOUNTS * * *"
1310 INPUT"IS THE LINEPRINTER REQUIRED (Y/N) ";P$
1320 IFP$<>"Y"ANDP$<>"N"THEN1310
1330 INPUT"WHICH ACCOUNT NO DO YOU REQUIRE ?↑
     (ENTER '999' TO EXIT AND RETURN TO MENU) ";N
1350 IFN>=1ANDN<=100GOTO1370
1360 IFN=999THEN50ELSE1330
```

```
1370 PRINTF1$: IFP$="Y"THENLPRINTF1$
1380 FORI=1TO100: IFA(I)=""THEN1440
1390 IFN<>VAL(MID$(A(I),42,2))THEN1440
1400 PRINTA(I): IFP$="Y"LPRINTA(I)
1420 DR#=VAL(MID$(A(I),47,8)): CR#=VAL(MID$(A(I),55,8)): DT#=DT#+DR#:
     CT#=CT#+CR#: GT#=GT#+DR#-CR#
1440 NEXT
1450 PRINT"TOTALS.....";: PRINTTAB(38)USINGF4$;DT#;CT#
1460 IFP$="Y"LPRINT
     "TOTALS.....";: LPRINTTAB(38)USING F4$;DT#;CT#
1470 PRINT"BALANCE OF ACCOUNT.....";: PRINTTAB(38)USINGF4$;GT#
1480 IFP$="Y"LPRINT
     "BALANCE OF ACCOUNT.....";: LPRINTTAB(38)USING F4$;GT#
1490 GT#=0: DT#=0: CT#=0: PRINT: PRINT: IFP$="Y"LPRINT" ": LPRINT
     " "
1500 GOTO1330



 100 REM *** SUPER SIZZLER ***

 110 REM AUTHOR ROBERT A SUNNERS, 26 SUNCROFT ST
 120 REM MT. GRAVATT, BRISBANE 4122 - 07 349 2598
 125 REM MEM REQ 8K
 130 CLS: PRINT,
     "SUPER SIZZLER IS A GAME OF LIFE": PRINT,STRING$(31,"+"):
     PRINT: PRINT,
     "TO BUILD A VAN FROM THE GROUND UP YOU MUST FIRST MAKE SOME MONEY. USE
     THE ";CHR$(93);" AND ";CHR$(94);" KEYS ";
 140 PRINT"TO POSITION YOUR GUN ON  THE GAPS IN THE BARRIERS.":
     PRINT,"FIRE THE GUN BY PRESSING THE SPACER BAR. YOU CANMOVE THE GUN
     FASTER BY PRESSING THE @ KEY AFTER THE DIRECTION   HAS BEEN
     ESTABLISHED BY THE ARROW KEY."
 150 PRINT,"HITTING THE $ TARGET PAYS YOUR WAGE - HITTING   THE X TARGET
     WINS YOU A VAN PART FOR FREE. HITTING THE BARRIERS COSTS YOU MONEY.
     WHEN YOU HAVE SUFFICIENT CASH YOU WILL AUTO -  MATICALLY BUY A VAN
     PART."
 160 PRINT"HOWEVER, IF YOUR BILLS GET TOO HIGH YOU WILL BE FORCED TO SELL
     YOUR NEWEST PURCHASE.        << ENTER >>";:  INPUTA
 170 DATA"CRUNCH !!","WHUMP !!!","CRASH !!!","BANG !!!!","WHAMMY !!","SMASH
     !!!"
 180 DATAJACKS,CHASSIS,ENGINE,BODY,WHEELS,RADIO,TRIMS,VENTS,DOMES,TAKE OFF,
     50,750,650,2200,500,150,50,150,220,220
 190 CLS: DIMVL(12),PT$(12): IN=1: FORN=1TO6: READN$(N): NEXT:
     FORX=1TO10: READPT$(X): NEXT: FORX=1TO10: READVL(X): NEXT:
     V$="$$#####.##": BL$=
     "                ": PR=1: PS=-1: CP=VL(PR)
 200 PRINT@66,CHR$(173);CHR$(170);CHR$(149);CHR$(159);CHR$(170);CHR$(145);
     CHR$(159);" ";CHR$(136);CHR$(136);CHR$(136);
 210 PRINT@194,CHR$(173);CHR$(170);CHR$(168);CHR$(133);CHR$(158);CHR$(170);
     CHR$(144);CHR$(183);CHR$(170);CHR$(133);CHR$(144);
 220 FORY=0TO16STEP2: FORX=0TO1: SET(X,Y): SET(X+92,Y): SET(X+126,Y):
     SET(X,Y+31): SET(X+46,Y+31): SET(X+82,Y+31): SET(X+126,Y+31):
     NEXT: NEXT
 230 FORX=0TO4STEP2: PRINT@960+X,CHR$(179);: NEXT: FORX=19TO61STEP2:
     PRINT@960+X,CHR$(179);: NEXT: FORX=94TO125: SET(X,10): NEXT:
     FORX=84TO125: SET(X,37): NEXT
 240 PRINT@733,CHR$(183);CHR$(170);CHR$(170);CHR$(144);CHR$(181);CHR$(138);
     CHR$(148);
 250 FORX=0TO127: SET(X,0): SET(X,21): SET(X,23): NEXT: FORX=0TO127:
     SET(X,16): SET(X,31): NEXT: PRINT@384,"";: TG$=
     "$$$$$$$$$": FORX=1TO8: PRINTTG$;: NEXT
```

```
260 REM * SPARE PARTS SHOP *
270 FORX=6TO36STEP10: FORY=33TO36STEP3: SET(X,Y): SET(X+1,Y):
    SET(X+2,Y): SET(X+3,Y): NEXT: NEXT
280 FORX=4TO34STEP10: FORY=34TO35: SET(X,Y): SET(X+1,Y): SET(X+6,Y):
    SET(X+7,Y): NEXT: NEXT: IFXX=1THENRETURN
290 FORY=38TO43: SET(27,Y): NEXT
300 FORX=32TO41: SET(X,38): IFX>35ANDX<38THENX=38: NEXTELSENEXT:
    IFXX=1THENRETURN
310 PRINT@848,CHR$(48);: FORX=32TO36STEP2: SET(X,43): NEXT:
    FORX=37TO41STEP2: SET(X,40): SET(X,41): NEXT: IFXX=1THENRETURN
320 PRINT@304,"BANK";: PRINT@747,CHR$(94);CHR$(94);: PRINT@764,CHR$(93);
    CHR$(93);:  PRINT@749,
    "PARTS CATALOGUE";: PRINT@966,
    "PARTS  STORE";: PRINT@112,
    "GAIN";: PRINT@176,
    "LOSS";: GOSUB1200: PS=0
500 REM * TARGET FIRING OPERATION *
510 GN=606: GG=60: RANDOM: FORX=1TO5: P=RND(63): PRINT@384+P,"X";:
    NEXT
520 GH=30: PRINT@GN,
    "  [ ";: FORM=GGTOGG+6: SET(M,GH): NEXT: FORY=1TO10: A(Y)=RND(126):
    B(Y)=RND(127): NEXT
530 FORX=0TO127: FORY=1TO10: IFA(Y)>124THENTA=A(Y): A(Y)=0:
    RESET(A(Y)+2,21): SET(TA,21): SET(TA+1,21)ELSERESET(A(Y)+2,21):
    RESET(A(Y)+1,21): SET(A(Y),21): TX=X: TY=Y: GOSUB560  :
    X=TX: Y=TY
540 IFB(Y)<3THENTB=B(Y): B(Y)=127: RESET(B(Y)-2,23): SET(TB,23):
    SET(TB-1,23)ELSERESET(B(Y)-2,23): RESET(B(Y)-1,23): SET(B(Y),23)
550 A(Y)=A(Y)+1: B(Y)=B(Y)-1: NEXT: NEXT: GOTO530
560 REM * FIRING SYSTEM *
570 A$="": A$=INKEY$: IFA$=""THENRETURNELSEIF(A$<>CHR$(8))AND(A$<>CHR$(9)
    ANDA$<>CHR$(64))THENGOSUB610  :  RETURN
580 IFA$=CHR$(8)THENDR$="L": GOTO590  ELSEIFA$=CHR$(9)THENDR$="R":
    GOTO600  ELSEIFA$=CHR$(64)THEN650
590 IFGG>2THENPRINT@GN-1," [ ";: SET(GG-2,30): SET(GG-1,30):
    RESET(GG+6,30): RESET(GG+5,30): GG=GG-2: GN=GN-1: RETURNELSERETURN
600 IFGG<119THENPRINT@GN+1," [ ";: SET(GG+8,30): SET(GG+7,30):
    RESET(GG,30): RESET(GG+1,30): GG=GG+2: GN=GN+1: RETURNELSERETURN
610 IFA$=CHR$(32)THEN620  ELSEIFA$=CHR$(64)THEN650  ELSERETURN
620 M=PEEK(GN+15169): FP=GG+3: FORD=26TO19STEP-1: SET(FP,D):
    RESET(FP,D+1): IFPOINT(FP,D-1)=-1THEN630  ELSEIFD=19AND(M=36)OR(M=88)
    THENNEXT:  GOTO640  ELSENEXT: RETURN
630 PRINT@GN," [ ";: RESET(FP,D): PRINT@GN-62,"";: N=RND(6):
    PRINTN$(N);: FORV=1TO100: NEXT: GOSUB700  : PRINT@GN-62,
    "          ";: RETURN
640 PRINT@GN," [ ";: IFM=36THENPRINT@GN-191,CHR$(36);: GOSUB670:
    WW=RND(30000)/100: PRINT@117,USINGV$;WW;: PRINT@182,
    "          ";: GOSUB770  : RETURNELSEPRINT@GN-191,CHR$(88);:
    GOSUB670: PS=1: TT=TT+CP: WW=0: GOSUB720  : RETURN
650 IFDR$="L"THEND$=INKEY$: IFD$=""THENGOSUB590  : GOTO650  ELSERETURN
660 IFDR$="R"THEND$=INKEY$: IFD$=""THENGOSUB600  : GOTO660  ELSERETURN
670 PRINT@GN,"  ";: FORM=GGTOGG+6: RESET(M,30): NEXT: GG=60:
    GN=606: PRINT@GN,
    "  [ ";: FORM=GGTOGG+6: SET(M,GH): NEXT: RETURN
700 REM * BILL CALCULATIONS *
710 BL=RND(20000)/100+50: PRINT@181,USINGV$;BL;: PRINT@117,
    "          ";: TT=TT-BL
720 PRINT@309,USINGV$;TT;: IFPS<>0THENGOSUB800: PRINT@GN-62,
    "GO !";: FORX=1TO10: NEXT: PRINT@GN-62,
    "          ";
```

```
730 DB=-TT: NB=INT(DB/100): FORT=793TO921STEP64: FORY=0TO14:
    PRINT@T+Y," ";: NEXT: NEXT: IF-DB>0THENRETURNELSEIFNB>12THENNB=12
740 S=1: V=0: FORB=1TONB: V=V+1: FORX=43+V*9TO50+V*9: SET(X,45-S*2):
    NEXT: IFV=3THENV=0: S=S+1: NEXTELSENEXT
750 IFNB=12THENGOSUB780: GOTO720ELSEPRINT@GN-62,
    "GO !";: FORX=1TO10: NEXT: PRINT@GN-62,
    "      ";: RETURN
770 TT=TT+WW: IFTT>=CPTHENPS=1: GOTO720ELSEPS=0: GOTO720
780 FORY=1TO10: PRINT@939,
    "** PART FOR SALE **";: FORX=1TO100: NEXT: PRINT@939,STRING$(19,"*");:
    FORX=1TO50: NEXT: NEXT: PRINT@939,STRING$(19," ");: PS=-1:
    PR=PR-1: RETURN
800 REM * CONSTRUCTION PRIORITIES *
810 IFPS=-1ANDPR<1THENGOTO2100ELSEONPRGOSUB1000,1100,1200,1300,1400,1500,
    1600,1700,1800,2000
820 IFPS=-1THENTT=TT+VL(PR): CP=VL(PR)ELSETT=TT-VL(PR): PR=PR+1:
    CP=VL(PR)
830 PRINT@875,PT$(PR);STRING$(10-LEN(PT$(PR)),".");USINGV$;VL(PR);:
    PRINT@939,PT$(PR+1);STRING$(10-LEN(PT$(PR+1)),".");USINGV$;VL(PR+1);:
    PS=0
840 FORX=1TO10: PRINT@309,
    "* TRADE *";: FORY=1TO50: NEXT: PRINT@309,USINGV$;TT;: FORY=1TO50:
    NEXT: NEXT: RETURN
1000 REM * JACKS *
1010 IFPS=-1THEN1020 ELSEX=41: FORY=10TO13: SET(X,Y): SET(X+34,Y):
     SET(X+1,Y): SET(X+35,Y): IFY=13SET(X-1,Y): SET(X+2,Y): SET(X+33,Y):
     SET(X+36,Y): RETURNELSENEXT: RETURN
1020 X=41: FORY=10TO13: RESET(X,Y): RESET(X+34,Y): RESET(X+1,Y):
     RESET(X+35,Y): IFY=13RESET(X-1,Y): RESET(X+2,Y): RESET(X+33,Y):
     RESET(X+36,Y): RETURNELSENEXT: RETURN
1100 REM * CHASSIS *
1110 IFPS=-1THEN1130 ELSEFORX=34TO86: IF(X<40)OR((X>43ANDX<74))OR(X>77)
     THENSET(X,11): NEXTELSENEXT
1120 FORX=38TO79: IF(X<46)OR(X>71)THENSET(X,10): NEXT: RETURNELSENEXT:
     RETURN
1130 FORX=208TO237: PRINT@X," ";: GOSUB1010 : RETURN
1200 REM * ENGINE *
1210 FORY=1TO3: X=0: GOSUB1240 : NEXT: FORX=1TO18: Y=2: GOSUB1240 :
     NEXT: FORX=3TO15: Y=1: GOSUB1240 : NEXT
1220 FORX=4TO13: FORY=0TO3STEP3: IFY=3ANDX=4THENNEXT: NEXTELSEGOSUB1240 :
     NEXT: NEXT
1230 IFIN=0THENFORX=40TO44: SET(X,10): NEXT: RETURNELSEIN=0:
     RETURN
1240 IFPS=-1THENVX=35+X: VY=7+Y: RESET(VX,VY): PX=4+X: PY=38+Y:
     SET(PX,PY): RETURN
1250 VX=35+X: VY=7+Y: SET(VX,VY): PX=4+X: PY=38+Y: RESET(PX,PY):
     RETURN
1300 REM * DRAW VAN OUTLINE *
1310 IFPS=-1THEN1370 ELSEFORX=56TO85: FORY=4TO6: SET(X,Y): NEXT:
     NEXT: SET(55,5): FORX=54TO86STEP32: SET(X,6): NEXT: SET(55,6)
1320 FORX=34TO87: SET(X,7): NEXT: FORX=32TO88: FORY=8TO10: SET(X,Y):
     NEXT: NEXT: FORX=49TO70: SET(X,11): NEXT
1330 FORX=39TO73STEP34: RESET(X,9): RESET(X+1,9): RESET(X+2,9):
     RESET(X+3,9): RESET(X+4,9): NEXT
1340 FORX=37TO45: RESET(X,10): NEXT: FORX=72TO79: RESET(X,10):
     NEXT: FORX=34TO36: SET(X,11): NEXT: FORX=81TO86: SET(X,11):
     NEXT
1350 FORX=37TO48: RESET(X,11): NEXT: FORX=71TO80: RESET(X,11):
     NEXT: FORX=41TO75STEP34: FORY=9TO11: SET(X,Y): SET(X+1,Y):
     NEXT: NEXT
1360 FORX=57TO65: RESET(X,5): NEXT: FORX=56TO65: RESET(X,6):
     NEXT: FORX=55TO65: RESET(X,7): NEXT: RETURN
```

```
1370 FORX=15438TO15630STEP64: FORM=1TO31: POKEX+M,32: NEXT: NEXT:
     PS=1: GOSUB1000 : GOSUB1100 : GOSUB1200 : RETURN
1400 REM * WHEELS *
1410 IFPS=-1THEN1450 ELSEFORX=16066TO16086: POKEX,32: NEXT: FORX=6TO39:
     RESET(X,36): NEXT
1420 FORX=40TO74STEP34: FORY=10TO13STEP3: SET(X,Y): SET(X+1,Y):
     SET(X+2,Y): SET(X+3,Y): NEXT: NEXT: FORX=40TO74STEP34: FORZ=1TO4:
     FORY=11TO12: RESET(X+Z,Y): NEXT: NEXT: NEXT
1430 FORX=38TO72STEP34: FORY=11TO12: SET(X,Y): SET(X+1,Y): SET(X+6,Y):
     SET(X+7,Y): NEXT: NEXT
1440 FORX=39TO73STEP34: FORY=1TO5: RESET(X+Y,9): NEXT: NEXT:
     RETURN
1450 FORX=38TO72STEP34: FORZ=0TO7: FORY=10TO13: RESET(X+Z,Y):
     NEXT: NEXT: NEXT: PS=1: GOSUB1000 : XX=1: GOSUB270  : XX=0:
     RETURN
1500 REM * RADIO *
1510 IFPS=-1THEN1520 ELSEFORY=0TO5: RESET(27,38+Y): SET(50,1+Y):
     NEXT: RETURN
1520 FORY=0TO5: RESET(50,1+Y): SET(27,38+Y): NEXT: RETURN
1600 REM * ORNAMENT *
1610 IFPS=-1THEN1620 ELSEPRINT@144,CHR$(48);: PRINT@848," ";:
     FORX=30TO86STEP56: FORY=0TO4STEP2: SET(X+Y,10): RESET(X+Y+1,10):
     NEXT: NEXT: FORX=32TO41: RESET(X,38): NEXT: RETURN
1620 PRINT@144,CHR$(188);: PRINT@848,CHR$(48);: FORX=30TO90STEP60:
     RESET(X,10): NEXT: SET(33,10): SET(35,10): SET(87,10): XX=1:
     GOSUB300  : XX=0: RETURN
1700 REM * VENTS *
1710 IFPS=-1THEN1720 ELSEFORX=912TO914: PRINT@X," ";: NEXT: FORX=850TO852:
     PRINT@X," ";: NEXT: FORX=47TO51STEP2: RESET(X,9): NEXT:
     FORX=68TO72STEP2: FORY=6TO7: RESET(X,Y): NEXT: NEXT: RETURN
1720 FORX=47TO51STEP2: SET(X,9): NEXT: FORX=68TO72STEP2: FORY=6TO7:
     SET(X,Y): NEXT: NEXT: XX=1: GOSUB310  : XX=0: RETURN
1800 REM * WINDOW *
1810 IFPS=-1THEN1820 ELSEFORX=76TO83: FORY=6TO7: RESET(X,Y):
     NEXT: NEXT: RETURN
1820 FORX=76TO83: FORY=6TO7: SET(X,Y): NEXT: NEXT: RETURN
2000 FORX=0TO127: RESET(X,0): NEXT: FORX=0TO2: FORY=46TO302STEP64:
     PRINT@X+Y," ";: NEXT: NEXT
2010 FORX=0TO14: FORY=14TO270STEP64: PRINT@Y-X," ";: PRINT@Y+35+X," ";:
     NEXT: NEXT: AA=1: BB=15
2020 FORX=0TO127: SET(X,14): NEXT: FORX=0TO21: FORY=874TO938STEP64:
     PRINT@X+Y," ";: NEXT: NEXT: PRINT@878,
     "# # Y O U # #";: PRINT@942,
     "# # W I N # #";
2030 PRINT@278,CHR$(179);: PRINT@295,CHR$(179);: PRINT@211,CHR$(179);:
     PRINT@228,CHR$(177);: PRINT@213,CHR$(140);: PRINT@230,CHR$(140);:
     PRINT@276,CHR$(188);: PRINT@293,CHR$(188);: X=192: Y=256
2040 PRINT@X+20,CHR$(32);: PRINT@X+22,CHR$(130);: PRINT@X+37,CHR$(32);:
     PRINT@X+39,CHR$(131);: PRINT@Y+19,CHR$(176);: PRINT@Y+21,CHR$(176);:
     PRINT@Y+36,CHR$(176);: PRINT@Y+38,CHR$(176);
2050 PRINT@X+39,CHR$(179);: PRINT@Y+39,CHR$(176);: PRINT@X+22,CHR$(178);:
     PRINT@Y+22,CHR$(176);: PRINT@X+36,CHR$(129);: PRINT@Y+36,CHR$(179);
2060 PRINT@X+19,CHR$(131);: PRINT@Y+19,CHR$(179);: PRINT@X+38,CHR$(32);:
     PRINT@X+37,CHR$(140);: PRINT@Y+37,CHR$(176);: PRINT@Y+38,CHR$(188);
2070 PRINT@X+21,CHR$(32);: PRINT@X+20,CHR$(140);: PRINT@Y+20,CHR$(176);:
     PRINT@Y+21,CHR$(188);: FORA=AATOBBSTEP3: PRINT@X+45+A," ";:
     PRINT@X+46+A,"0";: PRINT@X+47+A,"  ";: NEXT: AA=AA+1: BB=BB+1:
     IFAA=3THENAA=0: BB=BB-3
2080 GOTO2030
```

```
2100 CLS: PRINTCHR$(23): PRINT: PRINT
2110 PRINT"    YOUR BILLS HAVE EXCEEDED↑
        YOUR CAPACITY TO REPAY.↑
        YOU ARE INSOLVENT !!!!!"
2120 PRINT: PRINT
     "    PLAY AGAIN ? (Y/N)  ";
2130 A$="": A$=INKEY$: IFA$=""THEN2130 ELSEIF(A$<>"Y")AND(A$<>"N")THEN2100
     ELSEIFA$="Y"THENRUN
```

```
  1 REM ** PROGRAM TO JOIN TWO CO-ORDINATES **

  2 REM ** PROGRAM NAME "CONNECTX/BAS" ..... **
  3 REM ** AUTHOR ROBERT A SUNNERS, 26 SUNCROFT ST.,
  4 REM ** MT. GRAVATT, BRISBANE 4122 07 349 2598
  5 CLS
100 INPUTX1
110 INPUTY1
120 INPUTX2
130 INPUTY2
150 IFABS(X2-X1)>ABS(Y2-Y1)THEN170
160 ST=SGN(Y2-Y1): X=X1: FORY=Y1TOY2STEPST: X=X+(X2-X1)/ABS((Y2-Y1)):
    SET(X,Y): NEXT: GOTO180
170 ST=SGN(X2-X1): Y=Y1: FORX=X1TOX2STEPST: Y=Y+(Y2-Y1)/ABS((X2-X1)):
    SET(X,Y): NEXT
180 GOTO180
```

```
  1 REM ** PROGRAM TO JOIN TWO CO-ORDINATES **
  2 REM ** PROGRAM NAME "CONNECTA/BAS" .... **
  3 REM ** AUTHOR ROBERT A SUNNERS, 26 SUNCROFT ST
  4 REM ** MT. GRAVATT, BRISBANE 4122 07 349 2598
  5 CLS
100 INPUTX1: X1=INT(X1/2)
110 INPUTY1: Y1=INT(Y1/3)
120 INPUTX2: X2=INT(X2/2)
130 INPUTY2: Y2=INT(Y2/3)
150 IFABS(X2-X1)>ABS(Y2-Y1)THEN170
160 ST=SGN(Y2-Y1): X=X1: FORY=Y1TOY2STEPST: X=X+(X2-X1)/ABS((Y2-Y1)):
    PRINT@X+INT(Y+.5)*64,CHR$(42);: NEXT: GOTO180
170 ST=SGN(X2-X1): Y=Y1: FORX=X1TOX2STEPST: Y=Y+(Y2-Y1)/ABS((X2-X1)):
    PRINT@X+INT(Y+.5)*64,CHR$(43);: NEXT
180 GOTO180
```

(In view of the high proportion of readers who are now subscribing to the monthly cassette edition, the tape index will be included as a regular part of the Magazine.)

-=*** FOR OUR CASSETTE SUBSCRIBERS ***=-

ISSUE 6 MAY 1980

All this month's programs, with the exception of Submarine Attack, have been recorded twice. All Level I programs are on side 1, and all Level II programs are on side 2.

PROGRAM	ID	approx start CTR41	CTR80
Space Drive L1/16K		8	5
		217	135
Sub Attack L1/4K		387	240
Super Sizzler L2/16K	S	21	13
		105	65
Household Accnts L2/16K	H	185	115
		242	150
Tic Tac Toe L2/4K	T	300	186
		326	202
Trig/Bas L2/4K	G	354	220
		382	237

---

tick where appropriate

To MICRO-80
Please consider the enclosed program for...

(i) Publication in MICRO-80

(ii) Publication on disk or cassette only

(iii) Both

Name.....

Address.....

.....

.....Post Code.....

***** CHECK LIST *****

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes seperated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it...etc

For system tapes, the start, end, and entry points, etc

The changes or improvements that you think may improve it.

Please package securely - padabags are suggested - and enclose stamps or postage if you want your cassette or disk returned.

We at MICRO-80 are pretty proud of our record. This is our sixth issue. We are half a year old, and have already become the leading software magazine in Australia. We have published over 40 major pieces of software and four hardware projects, at a cost of $12. (Half of the annual subscription). Our circulation is growing faster than ever. We have created avenues for our readers' to make their hobby profitable, where none existed before. We have created a sister company, MICRO-80 PRODUCTS, to further the interests of our readers, bringing them quality software and '80 add-ons at fair prices, and returning a fair portion of the proceeds of this venture to the readers who author the material sold. We are determined to go on to bigger and better things in the future, convinced that the TRS-80 and the System 80 are the best personal computers available, both in computing power, and in simple dollar value.

DEC '79
Super Mastermind
Digital Clock
Snake
Merge
Loader
Word processor
Cassette Mod.

JAN '80
Ricochet
Frustration
Hangman
Life
Inventory
Monitor in BASIC

FEB '80
Hangman mk II
Amazin
Biorythm
Files
Set 2
Light Pen
New Monitor in BASIC

MAR '80
Learning Nim
Lunar Lander
Ricochet mk II
Invaders
Random number tests
Floppy Disk mod.

APRIL '80
Horse race
One armed bandit
Bandit
Krazy-Kat
Movie-maker
Lower case enable

* Coupon prices are inclusive of post & packing *

To: MICRO-80, P.O. Box 213, Goodwood, S.A. 5034.

Please rush me the items checked below.

.....	12 month subscription to MICRO-80 and my free software catalogue.	$24.00
.....	12 month subscription to MICRO-80 and the cassette edition, plus my free software catalogue.	$60.00
.....	The latest issue of MICRO-80.	$ 2.50
.....	Touchtype.	$20.45
.....	RPN Calculator	$25.45
.....	BMON	$20.45
.....	Dalek Chase	$15.50
.....	Learning Level II	$18.15
.....	BOX of 10 * SCOTCH C10 Tapes	$24.00
.....	Box of 10 * SCOTCH C30 Tapes	$28.00
.....	Lower Case Mod.	$ 6.00
.....	Me Think it MICRO-mazing v.1	$15.50
.....	Me Think it MICRO-mazing v.2	$15.50
.....	UBOAT	$15.50
.....	Space Invaders	$ 9.00
.....	TRS-80 ROM Handbook	$26.00

----------TOTAL ENCLOSED $.....

Please debit my Bankcard $.....

Number.....Expires.....

Name.....

Address.....

.....

.....Post Code.....

Bankcard

RYMAC SOFTWARE

I S N O W T H E S O L E

QUEENSLAND DISTRIBUTOR

F O R

MICRO-80

single issues

subscriptions

MICRO-80 products

DEALER ENQUIRIES WELCOME

FOR FURTHER INFORMATION

PHONE
GARY MC GREGOR
OR TONY MELIUS
ON (07) 376-3191

OR WRITE TO
RYMAC SOFTWARE
P.O. BOX 69
SPRING HILL 4000

MICRO-80

LEVEL II ROM REFERENCE MANUAL

by Edwin Paay

Published by MICRO-80 PRODUCTS

Written by Eddy Paay, the LEVEL II ROM REFERENCE MANUAL is the most complete explanation of the Level II BASIC interpreter ever published.

Part 1 lists all the useful and usable ROM routines, describes their functions explains how to use them in your own machine language programs and notes the effect of each on the various Z 80 registers.

Part 1 also details the contents of system RAM and shows you how to intercept BASIC routines as they pass through system RAM. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory—the only restriction is your own imagination!

Part 2 gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements, etc. It also describes the various formats used for BASIC, SYSTEM and EDITOR/ASSEMBLER tapes. Each section is illustrated by sample programs which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

The LEVEL II ROM REFERENCE MANUAL is intended to be used by machine language programmers. It assumes a basic understanding of the Z 80 instruction set and some experience of Assembly Language programming. But BASIC programmers too will benefit from reading it. They will gain a much better insight into the functioning of the interpreter which should help them to write faster, more concise BASIC programs.

MICRO-80